Inspekcje produktów informatycznych

miesięcznik profesjonalistów

11
1997

informatyka

Cena: 5,5 zł. ISSN 0542 9951 WYDAWNICTWO SIGMA NOT

Pismo informatyczne - ukazuje się od 1965 roku

P 1877/97

Szacowanie zasobów w projektach informatycznych

Obiektowość w bazach danych
koncepcje, nadzieje i fakty

Osadzanie obiektów multimedialnych w bazach danych

SYSTEMY TRWAŁEJ ARCHIWIZACJI DOKUMENTÓW I DANYCH

Oferujemy ekonomiczne, sprawne w obsłudze systemy archiwizacji, zarządzania i wyszukiwania informacji tworzące rodzinę rozwiązań

mis ARCHI+

- Tworzenie efektywnego i łatwego w obsłudze archiwum na trwałych nośnikach CD-R, co zapewnia absolutne bezpieczeństwo danych. Jest to szczególnie ważne w przypadku danych finansowo-księgowych. Zapis na nośnikach CDR jest zgodny z Ustawą o Rachunkowości z 1994 r.
- Wysoka efektywność wyszukiwania dokumentów w obrębie dużego archiwum, składającego się z kilkuset płytek oraz natychmiastowy dostęp do całego archiwum wszystkim użytkownikom sieci, niezależny od fizycznego położenia poszczególnych nośników.
- Dane zgromadzone w archiwum są skompresowane, dzięki czemu na 1 nośniku CD-R można zgromadzić aż 3 GB raportów ASCII.
- Możliwość elastycznego opisu archiwizowanych dokumentów za pomocą słów kluczowych, wartości, dat i innych.
- Możliwość przeszukiwania pełnotekstowego, według słów kluczowych, tekstów pochodzących z funkcji OCR (czas wyszukiwania dowolnego dokumentu nie przekracza 3 sekund.
- Możliwość odczytu dowolnego nośnika wchodzącego w skład archiwum na dowolnym komputerze z systemem Windows.
- Aplikacja przeszukująca i odtwarzająca zawiera wbudowane przeglądarki dla formatów: Word, WordPerfect, 1-2-3, Excel, Quattro, PCX, TIFF, G3 i G4, DCX, GIF, BMP, Targa, EPS, WPG, PICT, JPEG, CALS raster, Sun raster, CLP, ICO, GEM, MSP, CAD, Kofax, GX2, MacPaint, RLE, Photoshop, LaseData, XBM, XPM, Brookout, HALO, IFF, IOC, XWD i inne.
- Systemy stworzone w oparciu o oprogramowanie firmy IMR stosowane są przez ponad 300.000 klientów na całym świecie (np. Boeing, US Navy, Black&Decker).

System składa się z następujących elementów:

- Stanowisko skanowania, tworzenia bazy, wyszukiwania i odtwarzania dokumentów:
  - skaner dokumentów (np. Fujitsu),
  - nagrywarka płyt CD-R (np. Yamaha),
  - system operacyjny wraz z wybranymi modułami oprogramowania firmy IMR.
- Stanowisko udostępniania archiwum dokumentów w sieci:
  - CD-ROM Serwer lub zmieniacz płyt CD (np. jukebox NSM)
  - system operacyjny wraz z oprogramowaniem zarządzającym nośnikami CD-R oraz udostępniającym archiwum w sieci (np. WinNT + IXOS)

ARCHI+
ZAKŁADY ENERGETYCZNE
Kompleksowy system archiwizacji raportów, rozliczeń z klientami, umów, zleceń, projektów, baz danych itp. dostosowany do specyfiki zakładów energetycznych

ARCHI+
FINANSOWO-KSIĘGOWY
Kompleksowy system tworzenia trwałego archiwum dokumentów finansowo-księgowych (zapis zgodny z Ustawą o Rachunkowości z 1994 r.)

ARCHI+
BIURA PROJEKTOWE
Kompleksowy system archiwizacji projektów CAD, dokumentacji, katalogów producentów, wykonawców, materiałów itp. w biurach projektowych

BĄDŹ PANEM SWEGO CZASU

mis
Management Informati
Systems Sp. z o.o

UWAGA! Nowy adres siedziby naszej firm

MIS Sp. z o.o., 51-502 Wrocław, ul. Mydlana 1, tel./fax (0-71) 3450110, 34501

informatyka

Miesięcznik
ISSN 0542-9951
INDEKS 36124

Ukazuje się elektroniczna wersja naszego miesięcznika. Oto nasz adres w Internecie: **www.pol.pl/informatyka/**

nr 11 1997
Listopad
Rok wydania XXXIII

REDAGUJE ZESPÓŁ:

dr Lesław **WAWRZONEK**
(redaktor naczelny)
mgr Zdzisław **ŻURAKOWSKI**
(redaktor)
mgr inż. Agnieszka **ZIÓŁKOWSKA**
(sekretarz redakcji)
Alina **KLEPACZ**
(redakcja techniczna, administracja)

KOLEGIUM REDAKCYJNE:

prof.dr hab. Leonard BOLC
mgr inż. Piotr FUGLEWICZ
prof. dr hab. Jan GOLIŃSKI
dr inż. Zenon KULPA
prof. dr inż. Jan MULAWKA
prof. dr hab. Wojciech OLEJNICZAK
mgr. inż. Jan RYŻKO
dr Witold STANISZKIS
dr inż. Jacek STOCHLAK
prof. dr hab. Maciej STOLARSKI
prof. dr hab. Zdzisław SZYJEWSKI
prof. dr hab. inż.
Ryszard TADEUSIEWICZ
prof. dr hab. Jan WĘGLARZ

PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab. Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

REDAKCJA:
00-950 Warszawa,
ul. Ratuszowa 11, p. 628
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159
e-mail: informat@pol.pl

PRENUMERATA:
tel. 40-35-86, 40-30-86
Prenumeratę przyjmujemy również w sieci INTERNET:
WWW.pol.pl./sigma_not
E-mail: kolpor.sigma@pol.pl

Materiałów nie zamówionych redakcja nie zwraca.

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach 3½" - w czystych kodach ASCII (Latin II, Mazovia) lub edytorach: Word 2, Word 6.
Po szczegółowe informacje dla Autorów prosimy zwracać się do redakcji.

Redakcja nie ingeruje w treść i formę ogłoszeń i innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

Ogłoszenia przyjmują:
- Redakcja, tel. 619-11-61
- Dział Reklamy i Marketingu
00-950 Warszawa, ul. Mazowiecka 12
tel.: 827-43-66, fax: 826-80-16

Okładka:
AGAT, Jerzy Burski i Andrzej Jacyszyn,
Łamanie:
Alina Klepacz, program PageMaker
Druk:
Drukarnia SIGMA NOT Sp. z o.o.

W numerze:

POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach
P1877/97

INFORMACJE

Dokąd zaprowadzi nas graficzny interfejs użytkownika? 3
Dennis Keeling 9

SYSTEMY INFORMACYJNE

Sieci rozległe z łączami prywatnymi, część II 10
Jan Piecha

Szacowanie zasobów w projektach informatycznych 15
Marek Miłosz

PUBLIKACJE

Obiektowość w bazach danych – koncepcje, nadzieje i fakty, część I 23
Kazimierz Subieta

Informatyka bankowa w kraju i na świecie, część II 27
Zygmunt Ryznar

Osadzanie obiektów multimedialnych w bazach danych 33
Piotr Porwik, Wojciech Jurczyk, Adam Trojan

Systemy czasu rzeczywistego

Inspekcje produktów informatycznych 37
Marek Jurkowlaniec, BjØrn Runge

BIULETYN PTI 46

W najbliższych numerach ukażą się artykuły:

System doradczy wspomagający projektowanie samotestowalnych układów cyfrowych VLSI – Dariusz Bojanowicz

Atrybuty wiarygodności systemów informatycznych wspierających prowadzenie ksiąg rachunkowych – Ignacy Dziedziczak

Obiekty funkcyjne w standardowej bibliotece języka C++ – Jarosław Jasiński

Podejście obiektowe w analizie bezpieczeństwa – Bartosz Nowicki, Steve Riddle

Systemy informatyczne dla banku uniwersalnego (cechy specyficzne) – Zygmunt Ryznar

Nowoczesne technologie zdobywania pieniędzy

Kiedy kilka miesięcy temu pytałem wpływowe osoby o informatyzację ZUS, otrzymywałem odpowiedzi w rodzaju „nie pchaj palców, gdzie tobie niemiło". Teraz już wiem, że chodziło o duże pieniądze, a informatyzacja była tylko pretekstem do ich zdobycia, bo przecież nie zarobienia.

Czy to nie dziwne, że podniósł się straszny hałas, gdy planowano zakup dla ZUS-u kilku samochodów, natomiast gdy na kilkanaście dni przed odejściem, rząd decyduje o wydaniu ok. 300 mln dolarów, panuje cisza? Za te pieniądze można polecieć dwa razy na Marsa lub kupić do każdego inspektoratu i oddziału ZUS-u (razem 300 jednostek) oprogramowanie i superkomputer Cray, taki, jaki jest na Politechnice Warszawskiej. Najszybszy komputer na świecie (9000 procesorów, 1000 GFLOPS) kosztował tylko 45 mln USD.

Nikt też nie zauważył, że przy okazji reformy systemu ubezpieczeń, ZUS funduje sobie komputerowy system ekonomiczny, a w tym moduły: finanse i księgowość, kadry i płace, gospodarka magazynowa i środki trwałe, inwestycje i wspomaganie zarządzania.

Anna Warchoł, rzecznik prasowy ZUS twierdzi, że przetarg został rozstrzygnięty zgodnie z prawem. A gdyby tak, szanowna Pani Rzecznik, był jeszcze przygotowany zgodnie ze zdrowym rozsądkiem i elementarną wiedzą z dziedziny realizacji dużych systemów informatycznych. Termin przeprowadzenia przetargu, sposób jego przygotowania i rozstrzygnięcia są – nie oszukujmy się – skandalem. Np. opis przedmiotu zamówienia zajmuje **sześć** stron maszynopisu (czcionka 14 pkt.). Wymagane referencje zostały tak skonstruowane, że do wskazania tego kto wygra potrzeba mniejszego wysiłku umysłowego, niż do odpowiedzi na jedno z telewizyjnych pytań, typu – W jakim filmie występował Roger Moore: Lalka, Seksmisja czy Święty?

Nowatorskim rozwiązaniem ZUS – czy aby zgodnym z ustawą o zamówieniach publicznych – jest organizacja przetargu bez zapewnienia źródeł finansowania kontraktu. Fundusz PHARE czy Bank Światowy nie wspomogą tego przedsięwzięcia, gdyż robią to wtedy, gdy przeprowadzają przetarg według własnych reguł, a więc rzetelnie.

Gdy czytam opracowanie „Kompleksowy System Informatyczny" (KSI ZUS), pewnie kwintesencja wiedzy 400 zatrudnionych tam informatyków, i uświadomię sobie, że na podstawie zawartych w nim dywagacji wydaje się trzysta milionów NASZYCH dolarów, to jak pisał poeta, ogrania mnie „śmiech pusty, a potem rozpacz i trwoga". Analiza techniczna porównująca różne modele budowy systemu (zdecentralizowany, pośredni i scentralizowany) jest infantylna, bo nie zasługuje na miano nieprofesjonalnej. Np. za modelem zdecentralizowanym przemawia fakt, że baza danych jednego inspektoratu jest mniejsza niż wspólna baza dla 250 inspektoratów lub że serwer jest tańszy od mainframe'a. Nie uwzględniono faktu, że roczny koszt stanowiska dla użytkownika mainframe'a wynosi ok. 1000 USD, natomiast 4000 USD w przypadku serwera UNIX w systemie z ponad tysiącem użytkowników (International Technology Group).

W opracowaniu tym nie wspomina się wcale, jak zostaną wykorzystane ogromne nakłady finansowe oraz ludzie z zakładów ZETO, obsługujący aktualnie system naliczania i wypłat emerytur. Co na przykład będzie z zakupionym w tym celu w czerwcu br. oprogramowaniem za kilka milionów DM?

Niewiele też wyjaśnia raport niemieckiej firmy GVG wykonany ze środków PHARE, nt. koncepcji reformy ZUS. Kosztował on prawie 2 mln USD i, po prawdzie, znam kilka osób, które zrobiłyby lepszy, nawet dziesięć razy taniej. Jest w nim wiele stwierdzeń tak ogólnikowych, że aż nieprawdziwych, a tam, gdzie potrzeba liczebników są przymiotniki. Nie przedstawiono wariantów informatyzacji ZUS-u, tak technicznych, jak i organizacyjnych z ich zaletami i wadami.

Katastrofa Poltaxu ma względnie skromne zasługi w odstraszaniu od technik informatycznych – mniej efektywnie ściągane są podatki i nie ma danych analitycznych, niezbędnych do prowadzenia mądrej polityki podatkowej. Państwo jest widać zasobne. Za to strach pomyśleć co będzie, gdy niezamożni emeryci nie dostaną w terminie swoich pieniędzy. Znowu będą winni informatycy, komputery i cykliści.

„*W październiku rozstrzygnięto przetarg na system informatyczny obsługi ZUS, choć nie są jeszcze znane nowe ustalenia w zakresie ubezpieczeń społecznych i emerytalnych. W przyszłości ZUS ma być zupełnie inną firmą o nieznanej jeszcze strukturze.*" – stwierdza Polska Izba Informatyki i Telekomunikacji. Jej członkowie zdają sobie sprawę, że jeszcze jeden poważny blamaż komputeryzacji, a społeczeństwo będzie przekonane, że inwestowanie w informatykę jest marnowaniem pieniędzy. Już teraz słychać opinie, że komputeryzacja przynosi największe oszczędności wtedy, kiedy się jej nie stosuje. Dlaczego Polskie Towarzystwo Informatyczne czy jego członkowie nie bronią przyszłości swoich miejsc pracy? Nie znana jest też opinia Rady Koordynacyjnej ds. Teleinformatyki, choć ocena takich projektów to jej statutowy obowiązek.

Za kilka lat, gdy okaże się, o czym jestem przekonany, że system informatyczny dla ZUS-u jest wielką klapą co do jakości, terminu lub/i kosztów, wszyscy, którzy teraz milczą nie będą mieli moralnego prawa wypowiadać się o tym, co i jak należało zrobić. Teraz jest jeszcze na to czas. Zapraszam do dyskusji na łamach „Informatyki".

Leszek Jaworowski

Kontrakt CSBI S.A.

Firma CSBI S.A. (BHT Wrocław) podpisała kontrakt z firmą Howell S.A. na sprzedaż systemu MFG/PRO – systemu zarządzania przedsiębiorstwem klasy MRP II. Wartość kontraktu oceniono na 181 tys. USD, w tym: 160 tys. USD to wartość systemu MFG/PRO i koszty wdrożenie, zaś 21 tys. USD to koszt komputera SUN Enterprise 150. Termin zakończenia realizacji kontraktu przewidywany jest na maj 1998. Będą wdrażana następujące elementy systemu: finanse, sprzedaż, zakupy, gospodarka magazynowa, zaś w przyszłości także moduły planowania, dystrybucji, planowania MRP i analizy finansowe i handlowe.

Pierwszy węzeł „Internetu dla Niepełnosprawnych" (IdN1)

W 1996 r. uczestnicy II Forum Teleinformatyki w Legionowie podjęli inicjatywę utworzenia „INTERNETU DLA NIEPEŁNOSPRAWNYCH". Jej głównym animatorem był zmarły tragicznie w br. Marek Car, Prezes Polskiej Społeczności Internetu oraz Prezes Polska OnLine. W grupie Inicjatorów znalazły się następujące organizacje i firmy: Telekomunikacja Polska S.A.,IBM Polska, Microsoft Polska Sp. z o.o., Polska OnLine, Zakład "Internet dla Szkół" Fundacji Rozwoju Demokracji Lokalnej oraz Fundacja Pomocy Matematykom i Informatykom Niesprawnym Ruchowo - Warszawa (FPMiINR), której Inicjatorzy powierzyli organizację pierwszego węzła „Internetu dla Niepełnosprawnych" – IdN1.

Już w fazie prac organizacyjnych idea IdN1 zyskała następnych sojuszników, którzy ją wsparli w znaczący sposób: Fundacja Stefana Batorego, Ambasada Kanady, Cisco Polska, Solidex, KARMA Polska International Group Sp. z o.o., TCH Components.

Dzięki wszystkim sponsorom – ludziom wielkiego serca – węzeł: idn1. idn.org.pl; www.idn.org.pl; 194.204.152.130 – rozpoczyna swoją działalność!

Kierownik Projektu IdN1
Krzysztof Markiewicz
e-mail: kmark@pol.pl
Zastępca Kierownika Projektu IdN1
Krystyna Karwicka-Rychlewicz
e-mail: karwicka@pol.pl.

System SPS POMOST

Na targach SOFTARG'97 Konsorcjum 2Si firma tworząca oprogramowanie dla projektu ALSO zaprezentowała swój system dla ośrodków pomocy społecznej SPS POMOST. Ekspozycja i seminarium towarzyszące cieszyły się dużym zainteresowaniem pracowników ośrodków pomocy społecznej z całej Polski, oczekujących na wdrożenie systemu POMOST. Ich wysoką ocenę zyskał przyjazny interfejs użytkownika oraz przyjęte rozwiązania technologiczno-merytoryczne. Ekspozycja była uzupełnieniem seminariów odbywających się na targach SOFTARG'97 prowadzonych przez pracowników projektu ALSO ze strony Ministerstwa Pracy i Polityki Socjalnej.

Porozumienie ITF Intertraffic z grupy Daimler Benz i Mera-Błonie

Podczas II Targów Komunikacji Miejskiej w Łodzi podpisane zostanie porozumienie między firmą **ITF Intertraffic,** należącą do podstawowej grupy przedsiębiorstw koncernu Daimler Benz, a polskim zakładem **Mera-Błonie**. Celem porozumienia jest utworzenie międzynarodowego konsorcjum z udziałem polskich zakładów produkcyjnych. Jego zadaniem będzie przygotowanie szerokiego zaplecza dla promocji i wdrażania nowych technologii w zarządzaniu komunikacją miejską.

System logistyczny dla małych przedsiębiorstw

Najnowszy produkt Instytutu Logistyki i Magazynowania w Poznaniu **System Informacji Towarowej (SIT)** adresowany jest do producentów, dystrybutorów i odbiorców towarów. Najważniejszym klientem SIT są firmy zaliczane do grupy małych i średnich przedsiębiorstw, które nie widzą ekonomicznego uzasadnienia budowania własnych ogólnopolskich systemów informatycznych.

Założeniem systemu jest gromadzenie, weryfikacja i obróbka danych logistycznych o towarach znajdujących się na terenie Polski. Dane te są udostępnianie w ogólnoświatowej sieci Internet (**www.sit.ilim.poznan.pl**), tak by były dostępne szybko i w dowolnym miejscu, również dla potencjalnych klientów spoza Polski.

System Informacji Towarowej finansowany jest ze środków Komitetu Badań Naukowych oraz Instytutu Logistyki i Magazynowania w Poznaniu.

Bank Przemysłowy S.A. w Łodzi kupuje system elektronicznej bankowości TeleKonto Plus.

Na mocy podpisanej ostatnio umowy generalnej pomiędzy Bankiem Przemysłowym S.A. w Łodzi a Suntech Sp. z o.o. system elektronicznej bankowości TeleKonto Plus zostanie wdrożony w całym banku. Obecnie system został wdrożony przez dwa oddziały i trwa podłączanie kolejnych klientów banku.

System TeleKonto Plus (Home/corporate Banking) jest pracującym w środowisku Windows programem umożliwiającym klientom banku pobieranie informacji bankowych i przekazywanie zleceń drogą telekomunikacyjną.

HP ułatwia instalację pakietu SAP R/3

Pod koniec sierpnia, podczas corocznej konferencji dla użytkowników oprogramowania SAP (SAPPHIRE '97), Hewlett-Packard zapowiedział dostarczanie klientom całościowego rozwiązania, na które składa się sprzęt, usługi oraz wstępnie konfigurowany, gotowy do działania w środowisku Windows NT system SAP Ready-to-Run R/3. Oferta HP dotyczy pakietu R/3 dla przedsiębiorstw, który działa na platformie intelowskich serwerów HP NetServer.

Nowa inicjatywa firmy SAP ma ułatwić i przyśpieszyć wdrażanie wdrażanie systemu R/3 w przedsiębiorstwach. Rozwiązanie oferowane przez HP, które zostało opracowane w ramach inicjatywy Ready-to-Run R/3, jest przeznaczone głównie dla systemów z 25 do 100 użytkowników.

Oddział Tandem w Polsce

W październiku firma Tandem Computers, jeden z producentów systemów super-

komputerowych, otworzyła oddział w Polsce. Dyrektorem oddziału jest Jarosław Mulewicz, absolwent SGH, który poprzednio pracował polskich filiach firm Digital Equipment Corporation i Sanchez Computers. W sierpniu br. nastąpiła fuzja firm Compaq i Tandem.

Firma Tandem specjalizuje się w dziedzinie kompleksowych rozwiązań dla sektora bankowego, ubezpieczeń, telekomunikacyjnego, przemysłowego, transportowego, oraz tych wszystkich, które wymagają systemów o najwyższej mocy i niezawodności.

W Polsce Tandem dostarczył system rezerwacji biletów PKP, obsługuje przedsiębiorstwo PolCard, które ma 30 tysięcy punktów sprzedaży. Tandem będzie działał w Polsce przede wszystkim w sektorze bankowym i telekomunikacyjnym. We wrześniu zawarł umowę na komputeryzację BGŻ, w ramach którego działa 1200 banków spółdzielczych.

Można oczekiwać, że tandem Compaq i Tandem będą w Polsce poważną konkurencją dla IBM, HP, Digital czy Optimus. Prawdopodobna jest też współpraca tych firm z największymi polskimi integratorami, w tym między innymi z Computerlandem i Apeximem.

Seminarium firm DIGITAL i SAP

25 września odbyło się wspólne seminarium firm Digital Equipment Polska i SAP Polska, na którym specjaliści obu firm zaprezentowali strategiczne metody zarządzania dla produkcji procesowej przy zastosowaniu zintegrowanych systemów wspomagających zarządzanie na przykładzie rozwiązań obu firm. Zwłaszcza położono nacisk na rozwiązania stosowane w przemyśle chemicznym i farmaceutycznym. Wspólne wdrożenia realizowane przez firmy SAP i DIGITAL zintegrowanego systemu zarządzania na przykładzie zakładów farmaceutycznych omówili goście przybyli z Węgier.

SAP R/3 współpracuje ono z większością znanych baz danych (bez Sybase) oraz platform sprzętowych i systemowych (IBM, HP, Sun, NRC, Siemens).

Procesor microJava 701

MicroJava 701 firmy Sun Microsystems to pierwszy procesor serii JavaChip wyspecjalizowany w wykonywaniu kodu rodzimego Javy. Procesor integruje funkcje systemowe w jednej kości, dzięki czemu eliminuje się potrzebę stosowania zewnętrznych układów ASIC. W sercu procesora microJava znajdzie się jądro, picoJava 2.0 zoptymalizowane dla wykonywania kodu Javy oraz kodu języka C. „Wykorzystując nasze możliwości projektowe i strategicznych partnerów, zamierzamy zaoferować całą rodzinę procesorów przeznaczonych dla Javy oraz licencje na jądra", stwierdził Raj Parekh Vice President i General Manager w Sun Microsystems Volume Products Group.

Procesor microJava 701 będzie wykorzystywał 32-bitową architekturę picoJava 2.0. Będzie zawierać jednostki obliczeniowe liczb stało- i zmiennoprzecinkowych oraz pamięci podręczne o wielkościach 16 dla instrukcji i 16KB dla danych. Procesor będzie produkowany

w technologii CMOS 0,25 mikrona, a docelowa prędkość jego zegara to 200MHz. Kontroler pamięci i szyny we/wy (I/O bus) zostały zintegrowane w urządzeniu. Projekt zakłada, że procesor będzie zasilany napięciem 2,5 volt (jądro) i 3,3 volt (I/O). Rozmiary procesora to 50 mm^2. Ogólna liczba tranzystorów wynosi ok. 2,8 mln. Procesor będzie posiadał obudowę (PBGA) z 316 pinami.

Sun przewiduje, że na skalę przemysłową procesor microJava 701 będzie produkowany w drugiej połowie 1998 roku.

Sieć szkieletowa ATM Microsoftu

Firmy FORE Systems i GTE zostały wybrane przez Microsoft do realizacji sieci szkieletowej oraz związanych z nią serwisów przy rozbudowie sieci kampusowej w swojej centrali w Redmont.

Sprzęt dla sieci w standardzie ATM dostarczy firma FORE Systems. Instalacja oraz usługi dodatkowe zostaną zrealizowane przez GTE. Sieć szkieletowa ATM Microsoftu, która stanie się jedną z największych instalacji tego typu na świecie, będzie m.in. obsługiwała internetowe centra danych (Internet Data Center) tej firmy w różnych krajach.

Produkt do składowania w buforze stron WWW

Cisco Cache Engine nowy produkt firmy Cisco Systems Inc. to pierwsze skalowalne urządzenie, które służy do lokalnego składowania danych w buforze i nie wymaga konfigurowania klienta, dzięki czemu jest niewidoczne dla użytkowników. Cisco Cache Engine pozwala zwiększać możliwości Internetu przez eliminowanie powtórnych transmisji tych samych stron WWW, umożliwia zapisywanie zawartości stron WWW lokalnie, dzięki czemu operatorzy sieci i duże firmy mogą zaoszczędzić na kosztach połączeń w sieciach rozległych (WAN).

Sun Microsystems i ComputerLand

W sierpniu 1997 roku firmy Sun Microsystems i ComputerLand Poland S.A., w następstwie kilkuletniej działalności firm na rynku produktów i usług sieciowych, podpisały porozumienie o wspólnym świadczeniu usług sieciowych i integracyjnych oraz pomocy technicznej dla klientów firm.

Centrum Obsługi Klienta Novella w Polsce

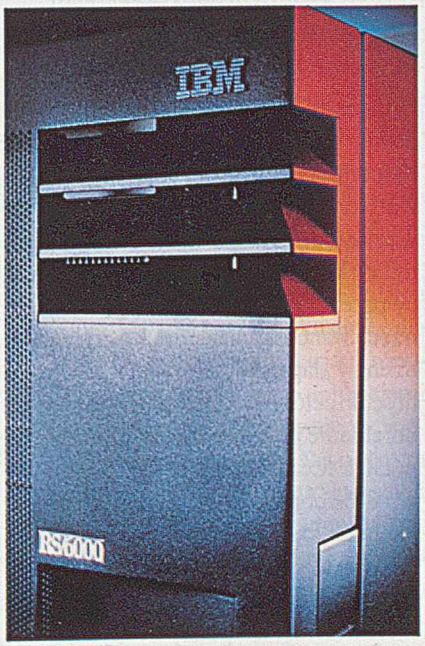
Głównym zadaniem Centrum Obsługi Klienta będzie pomaganie tym klientom, którzy planują inwestycje w technologie informatyczne i poszukują niezbędnych informacji. Centrum będzie więc pełniło funkcje doradcze, oferując pomoc w zakresie wyboru rozwiązań sieciowych opartych na produktach firmy Novell i sposobu realizacji zakupów.

Nowe produkty IBM

IBM przedstawił rodzinę stacji roboczych i serwerów IBM RS/6000 zaprojektowanych w celu zwiększenia wydajności w badaniach naukowych i technicznych oraz projektowaniu.

Do nowych produktów należą:

- Model 397: oparta na POWER2 SuperChip (P2SC) stacja robocza o największej wydajności obliczeń zmiennopozycyjnych;
- P2SC 160 MHz: Thin Node o zwiększonej wydajności dla RS/6000 SP;
- Rozszerzenia linii stacji roboczych 43P o procesor Model43P 140;
- POWER GXT 800M: wysokiej klasy karta graficzna realizująca funkcje grafiki trójwymiarowej;
- POWER GXT120P: nowa karta grafiki dwuwymiarowej;



Międzynarodowa konferencja SAP AG „Technology days" w Karlsruhe

SAP AG zorganizował w dniach 7-10 października 1997 r. w Karlsruhe międzynarodową konferencję „Technology days". Uczestniczyło w niej ok. 3000 osób, w tym przede wszystkim: specjaliści wdrażający system R/3, piszący dla niego aplikacje i użytkownicy.

Konferencji towarzyszyła wystawa firm współpracujących z SAP AG, w której uczestniczyli tacy potentaci rynku komputerowego, jak np.: IBM, Digital, Sun Microsystems, Siemens, HP, prezentowali oni platformy sprzętowe dla systemu R/3, zaś Microsoft, Lotus, Computer Associates, Oracle i Informix systemy zarządzania bazami danych i oprogramowanie uzupełniające i rozszerzające możliwości tego systemu, w tym przede wszystkim narzędzia do współpracy z internetem i intranetem oraz analizy zbiorów danych.

Na konferencję w Karlsrhue przybył też Bill Gates. W swoim wystąpieniu poświęconym głównie współpracy Microsoft Corp. z SAP AG przypomniał, że Microsoft jest zarządzany przy wykorzystaniu sytemu R/3 oczywiście na platformie Windows NT. Mówił też o udanej współpracy inżynierów obu firm nad opracowaniem kolejnych wersji oprogramowania do zarządzania przedsiębiorstwem, współpracującym z Windows NT. System zarządzania przedsiębiorstwem R/3 został zakupiony przez ponad 4000 klientów i ma 11500 instalacji, z tego 3400 na Windows NT, a 600 na SQL Server. Bill Gates przedstawił też słuchaczom swoją ideę Digital Nervous System, czyli objęcia całego świata jednym wielkim systemem informatycznym za pomocą łączy naziemnych i satelitarnych.

W konferencji prasowej, która obyła się podczas spotkania w Karlsruhe uczestniczyli członkowie rozszerzonego zarządu SAP AG: dr Peter Zencke, Gunther Tolkmit, Karl-Heinz Hess i rzecznik prasowy firmy Frank Schabel. Wśród dziennikarzy byli m.in. reprezentant miesięcznika Byte i InfoWorld, a także anali-

tycy Dennis Keeling i przedstawiciele Strategy Partners. Na pytanie „Dlaczego tak silnie związał się SAP z Microsoftem, Karl-Heinz Hess odparł, że zarówno standardów CORBA jaki i platform UNIX jest kilka i to stwarza jego firmie dodatkowe problemy, natomiast Windows NT jest jedno i choćby z tego powodu znacznie mniej kłopotliwe jest tworzenie systemu R/3 dla tej platformy.

Na konferencji prasowej padło wiele interesujących stwierdzeń i opinii. Gunther Tolkmit członek rozszerzonego zarządu SAP AG uważa, że: „Największym zagrożeniem w utrzymaniu czołowej pozycji na rynku oprogramowania biznesowego dla SAP AG jest sama firma, ponieważ znacznie trudniej jest nadawać samodzielnie tempo rozwoju w danej dziedzinie niż nadążać za konkurencją."

„Trzy decyzje miały decydujący wpływ na rynkowy i technologiczny sukces SAP AG. Podjęto je wtedy, kiedy nie były one takie oczywiste jak dziś. Były to: tworzenie oprogramowania w C a nie w Pascalu, związanie się z UNIX-em, kiedy jeszcze nie był produktem komercyjnym oraz wprowadzenie graficznego interfejsu użytkownika" – powiedział Karl-Heinz Hess członek rozszerzonego zarządu SAP AG.

Przed występem Billa Gatesa.

Uczestniczący w konferencji Dennis Keeling, angielski analityk oprogramowania dla biznesu przedstawił ranking firm, producentów oprogramowania do wspomagania zarządzania przedsiębiorstwem, według wielkości sprzedaży i usług z nim związanych. Według jego danych przoduje SAP AG z obrotami ponad 2,4 mld USD, za nim jest Oracle z szacowanymi na 1,2 mld USD obrotami i Computer Associates - 0,9 mld USD.

Na konferencji zaprezentowano nową wersję ABAP/4 wzbogaconą o oprogramowanie objektowe. Dotychczasowe aplikacje napisane w języku ABAP/4 będą pracowały z nowym językiem obiektowym ABAP bez konieczności modyfikacji istniejącego kodu. Język obiektowy ABAP będzie dostępny tylko jako część systemu R/3 i nie będzie oferowany jako standardowy język programowania.

Wnioski nasuwające z Technology Days to przede wszystkim to, że Business Framework dąży do modularyzacji, co oznacza, że nie będzie konieczności instalowania całego systemu R/3 w celu używania tylko części modułów, lecz będzie możliwość zainstalowania tylko tych, które będą używane. Pierwszym przykładem takiej strategii jest moduł HR /human resources/, który jest oferowany jako oddzielna aplikacja. Możliwość instalowania pojedynczych modułów powoduje decentralizację instalacji, łatwiejszy jest upgrade systemu (poszczególne moduły są niezależne od siebie i nie trzeba koniecznie upgradować całego systemu gdy modernizacja dotyczy pojedynczego modułu), możliwość instalacji systemów rozproszonych, możliwość kooperacji poszczególnych modułów z aplikacjami innych systemów spełniających standard BAPI /business application programing interfaces).

Zapowiedziano również, że od 01.99 będzie wersja SAP R/3 uwzględniająca nową jednostkę monetarną Euro. Oczywiście druga poważna zmiana czyli problem roku 2000 został już rozwiązany w oprogramowaniu SAP AG. (lw)

Przenośny komputer w szkole francuskiej

Francuskie Ministerstwo Edukacji Narodowej opublikowało wyniki programu eksperymentalnego dotyczącego wprowadzenia komputerów przenośnych w szkołach średnich. Projekt ten, nazwany „elektronicznym tornistrem", a trwający przez 5 lat (1991-95), objął uczniów trzech francuskich liceów, którzy, zaopatrzeni w „laptopy", testowali ich przydatność do celów dydaktycznych, takich jak odrabianie prac domowych czy robienie notatek w czasie zajęć.

80% objętych programem licealistów stwierdziło, iż komputer przenośny rzeczywiście stał się domową pomocą naukową. Nauczyciele zaś dostrzegli wzrost niezależności ich podopiecznych w wyniku eksperymentu. Postulowano lepsze dostosowanie parametrów komputerów do warunków szkolnych, m.in. zmniejszenie wagi czy też zwiększenie trwałości baterii.

Najważniejszym jednak wnioskiem z przedsięwzięcia jest to, iż użycie nowoczesnego narzędzia informatycznego nie miało wpływu na wyniki w nauce: słabi uczniowie pozostawali takimi nadal. W sprawozdaniu stwierdzono, że nie należy spodziewać się rozwiązania problemów w nauczaniu, których źródło leży głęboko w psychice, tylko przez wprowadzenie technicznych udoskonaleń do metodyki nauczania.

Agata Chrobot, *Paryż*

MiCOMP dla samorządów

W III kwartale 1997 roku firma MiCOMP wygrała przetargi w urzędach miast i gmin w Będzinie, Tychach, Mszanie, Dąbrowie Górniczej, Siemianowicach i Katowicach na oprogramowanie wspomagające pracę samorządów lokalnych. MiCOMP Systemy Komputerowe z Katowic specjalizuje się w kompleksowej komputeryzacji urzędów administracji samorządowej, która obejmuje dostawy sprzętu, wykonywanie okablowania strukturalnego, dostawy własnego oprogramowania aplikacyjnego wspomagającego zarządzanie miastem i gminą (**MAGISTRAT II for Windows**

Na SOFTARG'97 MiCOMP zaprezentował najnowszą wersję systemu MAGISTRAT II dla Windows. W wersji dla środowiska sieci lokalnej z systemem MS Windows NT lub Novell NetWare, z bazą danych MS SQLServer lub Centura SQLBase i Windows'95 na stacjach roboczych.

IFS Poland w hutach

IFS Poland podpisał z Zakładem Przetwórstwa Hutniczego „Stalprodukt" S.A. umowę na wdrożenie trzech modułów oprogramowania wspierających zarządzanie przedsiębiorstwem: IFS Dystrybucja, IFS Produkcja oraz IFS Finanse.

Stalprodukt, od niedawna obecny na Giełdzie Papierów Wartościowych, to jeden z najważniejszych polskich producentów blach transformatorowych,

prądnicowych, barier drogowych i kształtowników zimnogiętych.

Kontrakt będzie wkrótce rozszerzony o system IFS Remonty i IFS Dokumentacja. Już rozpoczęte prace wdrożeniowe trwać będą ok. półtora roku.

Konferencja INFOBAZY

Pierwsza krajowa konferencja INFOBAZY'97 – Bazy Danych dla Nauki odbyła się w czerwcu w Gdańsku, pod patronatem KBN. Zorganizowało ją Centrum Informatyczne TASK, Instytut Oceanologii PAN oraz Politechnika Gdańska. Była ona okazją do wymiany doświadczeń, przedstawienia prac oraz problemów, jakie występują podczas tworzenia zasobów informacji, dostępnych w naukowych sieciach komputerowych.

Tematyka konferencji, obejmowała zarówno aspekty prawne ochrony baz danych w systemie prawa autorskiego, narzędzia i technologie oraz zastosowania w rożnych dziedzinach nauki:

-bazy przyrodnicze
-bazy danych w gospodarce
-geograficzne systemy informacji GIS
-bazy chemiczne i techniczne
-medycyna i kultura
-bazy biblioteczne
-bazy informacji o nauce.

Wiele baz można było obejrzeć w czasie prezentacji, łącząc się z rożnymi ośrodkami w kraju. Odbywały się też prezentacje produktów firm komputerowych.

Planowane jest zorganizowanie podobnej konferencji w przyszłym roku lub za dwa lata.

IV Międzynarodowa Konferencja Naukowa „Metodyka zastosowań informatyki w dydaktyce i pracach badawczych

Konferencja odbyła się w dniach 11-12 września 1997 roku w Mrzeżynie. Celem konferencji było podzielenie się doświadczeniami naukowymi w zakresie zastosowań informatyki w dydaktyce i pracach badawczych. Tematyka zastosowań informatyki w dydaktyce dotyczyła głównie nowych metod, technik i środków kształcenia. Organizatorami konferencji były Akademia Rolnicza w Szczecinie, Wydział Ekonomiki i Organizacji Gospodarki Żywnościowej, Zakład Informatyki i Akademia Rolnicza w Poznaniu, Wydział Rolniczy, Katedra Metod Matematycznych i Statystycznych. Przewodniczącą Rady Programowej była prof. dr hab. Jadwiga Orylska.

Coś dla administratorów sieci

Novell Polska zaprezentował swoje nowe produkty na konferencji prasowej 21 października br. Pozwalają one, zdaniem przedstawicieli Novella, na tani, wydajny i przede wszystkim bezpieczny dostęp do internetu z sieci komputerowych. Do zaprezentowanych produktów należały:

- Novell Application Launcher 2.01, redukujący koszty administrowania oprogramowaniem dzięki zastosowaniu NDS i zautomatyzowaniu procesu dystrybucji aplikacji i uaktualnionych wersji programów do stacji roboczych pracujących pod Windows NT i innych systemów;
- IntranetWare Client do Windows NT wraz z Workstation Managerem, dzięki któremu można integrować serwery i domeny NT oraz administrować nimi;
- Novell Administrator do Windows NT (wersja 2.0c), pozwalający administratorom na wykorzystanie NDS do synchronizowania domen NT;
- GroupWise 5.2, z bogatymi możliwościami współdziałania i wbudowanymi funkcjami poczty elektronicznej do Internetu/Intranetu, dostępnymi na wszystkich platformach, włącznie z NT;
- MenageWise 2.5 – zintegrowany zestaw funkcji administracyjnych, które służą do całościowego monitorowania i kontrolowania środowisk sieciowych. Ułatwia on administrowanie serwerami NetWare/IntranetWare, analizowanie ruchu w sieci, automatyzowanie inwentaryzacji sieci, zdalne kontrolowanie oraz administrowanie siecią i poszczególnymi stacjami roboczymi, administrowanie aplikacjami sieciowymi oraz zapobieganie rozprzestrzenianiu się wirusów – wszystko za pomocą pojedynczej, centralnej stacji roboczej. Dodatkowy program MenageWise Agent for Windows NT Server umożliwia administrowanie serwerami Windows NT;
- BorderManager – umożliwia centralne zarządzanie dostępem użytkowników do informacji oraz zabezpieczenie i przyspieszenie tego dostępu na każdej „granicy" - punkcie, w którym łączą się dwie sieci. BorderManager ma mechanizmy kontroli dostępu na poziomie użytkowników, usługi wirtualnych sieci prywatnych, funkcje konwertowania adresów, mechanizmy rozsyłania oraz zdalnego dostępu.

System NDS (*Novell Directory Services*) do Windows NT uzupełnia strategię Novella zmierzającą w kierunku integracji z tym środowiskiem. System ten upraszcza i zabezpiecza procesy udostępniania zasobów w ramach jednej sieci na platformach serwerów NT i IntranetWare/NetWare, daje administratorom kompleksowe, zintegrowane narzędzie, za pomocą którego mogą administrować wszystkimi użytkownikami i zasobami przyłączonymi do sieci w środowisku NT i systemów Novella. (rm).

II Międzynarodowa Konferencja BIS '98

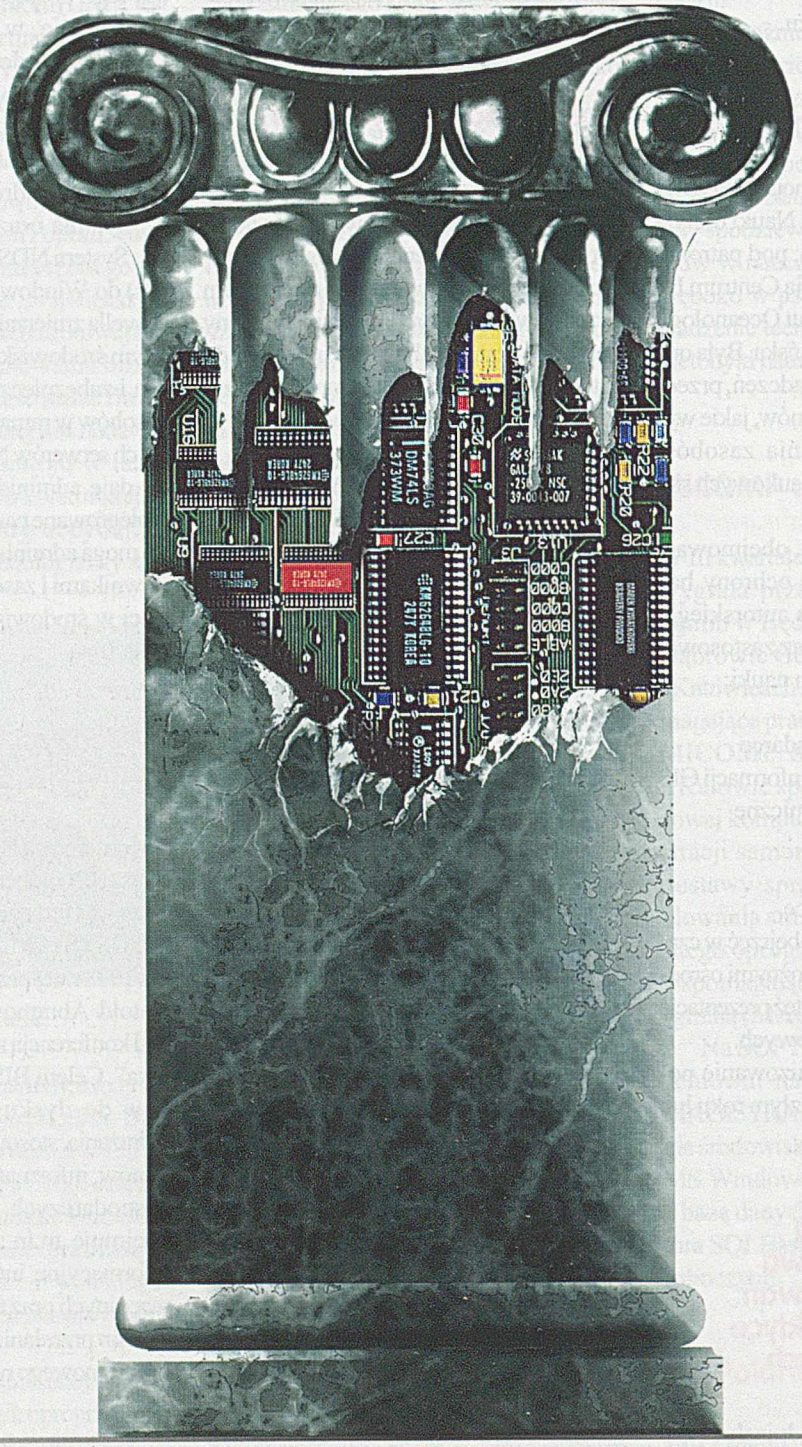
II Konferencja Business Information Systems (BIS) odbędzie się w Poznaniu podczas Międzynarodowych Targów INFOSYSTEM'98 w dniach 22 -24 kwietnia 1998 r. Jej przewodniczącym jest prof. Witold Abramowicz. Patronat prasowy nad konferencją sprawuje m.in. „Informatyka". Celem BIS jest stworzenie warunków do dyskusji na temat budowy, wdrażania, stosowania i doskonalenia systemów informatycznych dla procesów gospodarczych. Tematyka konferencji obejmuje m.in.: dziedzinowe systemy informacyjne, integrację systemów informacyjnych oraz transakcje elektroniczne. Termin przesłania artykułów do Komitetu Programowego mija 10 listopada 1997 r.

Informacje na temat konferencji BIS udostępniają:

Komitet Organizacyjny przy Akademii Ekonomicznej w Poznaniu, tel. (061) 869 92 61, fax (061) 866 89 24, e-mail bis@sylaba.poznan.pl, URL: http://www.bis.pozn.com.

Komitet Organizacyjny przy MPT: Biuro Informacji Technicznej – MPT, ul. Głogowska 14, 60-734 Poznań, tel. (061) 869 22 06, fax (061) 866 66 50.

UNOWOCZEŚNIAMY
TWÓJ BANK



SOFTBANK® S.A.

BANKOWOŚĆ I SYSTEMY BANKOWE

SOFTBANK S.A. ul.17-Stycznia 72A, 02-146 Warszawa, tel. 846 54 71, fax 846 14 57

Dokąd zaprowadzi nas graficzny interfejs użytkownika?

Standardowym interfejsem użytkownika dla aplikacji biznesowych jest Microsoft Windows, w wersji Windows 95 lub stacji roboczej Windows NT. Niestety graficzny interfejs użytkownika Windows nie poradził sobie z wymaganiami szybkiego dostępu do danych. Nie jest też rozsądną alternatywą dla milionów użytkowników terminali znakowych.

Przedsiębiorstwom bardzo trudno jest znaleźć uzasadnienie aby przejść do środowiska Windows opartego na komputerach PC, szczególnie że każda nowa wersja Windows wymaga wymiany PC, podczas gdy typowy prymitywny terminal znakowy przez ostatnich 15 lat był odpowiedni dla każdej następnej wersji oprogramowania.

Kolejną sprawą jest eksploatacja sieci PC działających pod Windows. W wielu badaniach wykazano niezbicie, że koszt utrzymania jednej stacji roboczej PC wynosi rocznie około 4500 funtów, a więc znacznie więcej niż prymitywnego terminala. Większość przodujących dystrybutorów oprogramowania twierdzi, że rozrost sieci klient/serwer ogranicza nie architektura, ale problemy z utrzymywaniem infrastruktury, która musi znajdować się wewnątrz korporacji i obsługiwać dużą i różnorodną grupę użytkowników.

Wielokrotnie próbowano, zahamować rozpowszechnianie się Microsoft Windows. Na początku lat 90-tych obóz UNIX-a, z firmą Sun Microsystems na czele, wylansował ideę UNIX-owej stacji roboczej, wyposażonej w Motif i X-windows. Świetny pomysł, ale nikt go nie kupił. Dziś ten sam Sun Microsystems stworzył zespół razem z uznanym autorytetem w dziedzinie baz danych Larry Ellison'em z Oracle, aby wylansować *komputer sieciowy*, czyli komputer PC pozbawiony stacji dysków, uzależniony od centralnego serwera. Komputer sieciowy może zastąpić prymitywny terminal, emulując jego tryb pracy i stanowiąc tanią alternatywę do komputerów PC. Jednak jak dotąd nie posiada on dobrego graficznego interfejsu użytkownika.

Komputer sieciowy może używać interfejsu przeglądarki sieciowej (ang. Web-browser). Przeglądarka sieciowa wymyślona przez Netscape, jako alternatywa do Windows, została obecnie zaadoptowana przez Microsoft i ich Internet Explorer stopniowo przełamuje dominację Netscape na tym rynku. Ale przeglądarka sieciowa nie jest alternatywą dla Windows, gdy idzie o szybki dostęp do danych. W rzeczywistości dostęp do danych za pomocą przeglądarki sieciowej nie jest przyjemny. Jak sama nazwa wskazuje, przeglądarka sieciowa doskonale nadaje się do przeglądania informacji, ale nie została zaprojektowana z myślą o dostępie do danych. Jest nieporęczna i nie nadaje się do pracy w skali przemysłowego dostępu do danych.

W 1996 roku, w odpowiedzi na zarzuty prasy o niezdolności do podążania w kierunku wyznaczonym przez przeglądarki sieciowe, SAP wprowadził swój własny interfejs dla aplikacji oparty na przeglądarkach sieciowych, opracowując ponad 120 aplikacji w ciągu następnego roku. Aplikacje te są wciąż w wersji Beta. SAP przyznaje, że rynek nie domagał się interfejsu sieciowego do rozpowszechnionych, działających w Windows aplikacji SAP R/3. Nawet Oracle, który był jednym z promotorów komputera sieciowego i przeglądarki sieciowej, a teraz wprowadził interfejs sieciowy do wszystkich swoich aplikacji, ekranów i raportów, przyznał pod koniec ostatniego roku, że bardzo niewielu użytkowników wybrało ten system jako alternatywę dla Microsoft Windows.

SAP ukierunkował przyszły rozwój R/3 na środowisko Windows. Nie zaoferował tekstowej alternatywy, co nie przeszkodziło mu w sukcesach. W 1996 roku stał się największą i najbardziej rozpowszechnioną aplikacją na świecie, uzyskując ze sprzedaży 2,4 miliarda dolarów. Z drugiej strony, Oracle miał duże problemy ze swoim produktem klient/serwer działającym w Windows i dlatego był zainteresowany w promowaniu alternatywnej, sieciowej aplikacji, która używa mocy obliczeniowej centralnego serwera. Zamiast ukierunkowania się tylko na Windows, jak planowano pod koniec roku 1996 w pierwotnej Wersji 11.0, Oracle odłożył tą ważną decyzję do końca 1997 roku. Teraz Wersja 11.0 będzie nie tylko Windows'owa, ale również sieciowa. Z powodu małego zainteresowania aplikacjami sieciowymi, jest mało prawdopodobne aby decyzja o wejściu w standard Windows zyskała uznanie wśród użytkowników terminali znakowych. Oracle ciągle oferuje znakowe wersje swoich większych implementacji.

Więc jaka jest alternatywa? Microsoft, zagrożony przez komputer sieciowy i problemy związane z utrzymaniem sieci PC z Windows, położył nacisk na rozwiój sieciowych PC. W ten sposób przenosi część aplikacji z PC z powrotem na serwer, aby zmniejszyć problemy związane z utrzymywaniem klienckich stacji roboczych PC. Przemysł przestawia się na aplikacje w Javie. Java ma swój własny interfejs użytkownika podobny do środowiska Windows, który jest niezależny od platformy i dostępny w formie Appletów. Nie wymaga on aby aplikacja znajdowała się u klienta. Eksperyment z Javą mogłyby doprowadzić do przełomu dla milionów użytkowników przykutych do prostych terminali znakowych.

Przyszłość nie jest jasna. Windows ma problemy z szybkim dostępem do danych i z utrzymaniem sieci, natomiast komputer sieciowy byłby odpowiedni, gdyby posiadał własny graficzny interfejs użytkownika. Przeglądarki sieciowe są wspaniałe do uzyskiwania informacji, ale nieporęczne przy dostępie do danych. Może Java jest rozwiązaniem z przyszłością.

Dennis Keeling

Sieci rozległe z łączami prywatnymi

Część II

Jan Piecha

Protokoły międzysieciowe – Internet

Grupy użytkowników komputerów współdziałają w realizacji określonych programów, których wyniki mogą zawierać interesujące informacje dla innych grup roboczych. Problemy transmisji danych na odległość są wtedy integralną częścią strategii rozwoju systemów projektowania „szkieletu" (backbone) rozległych sieci komputerowych. Transmisja danych w rozległych systemach komputerowych może odbywać się w dwojaki sposób:

- według protokołów połączeniowych, dla przesyłu niedużych plików;
- na podstawie protokołów bezpołączeniowych, dla transferu dużych wiadomości.

Transmisja według protokołów bezpołączeniowych ma swoje uzasadnienie w długim czasie trwania procesu transmisji, złożoności architektury sieci oraz w złożonym (wielostopniowym) sposobie adresacji. Zasady współdziałania odległych sieci lokalnych opisują protokoły komunikacji międzysieciowej, a współpraca tych sieci określana jest terminem internetworking lub w skrócie *internet*. Jednym z podstawowych systemów operacyjnych zarządzających pracą sieci „szkieletowych" jest Unix wyposażony w dwuwarstwowy protokół komunikacyjny TCP/IP.

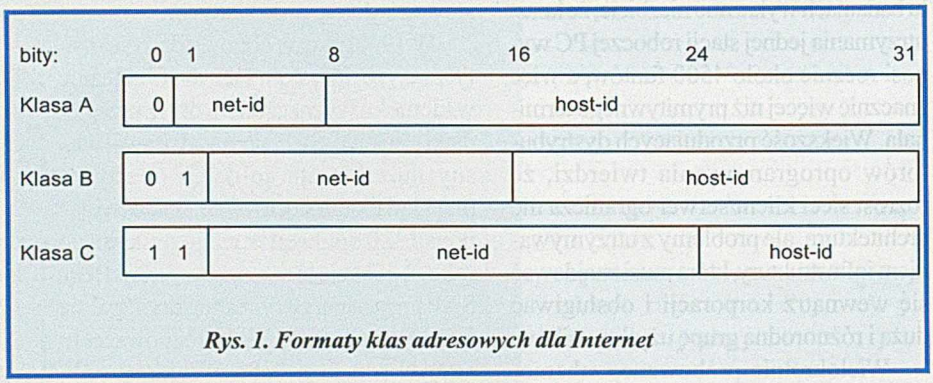
TCP/IP (Transport Control Protocol/ Internet Protocol) – odpowiada za połączenia międzywęzłowe sieci rozległych oraz przełączanie strumieni danych. Są to elementy organizacji procesu transmisji danych między węzłami lokalnymi, zainstalowanymi w rozległych sieciach komputerowych.

TCP/IP spełnia podobną rolę jak protokół IPX (Internet Packet eXchange) w sieciach lokalnych firmy Novell.

Adresacja

Węzły w sieci Internet posługują się 32-bitowymi adresami sieciowymi IP (trzecia warstwa modelu OSI-7). Zawierają one identyfikatory kolejnych przełączników (routingu) na drodze połączeń międzysieciowych. Każdy host komputer podłączony do sieci jest opisany 32-bitowym identyfikatorem, zwanym adresem sieciowym lub internetowym. Każdy adres zawiera parę identyfikatorów *net-id* (network-idetifier), będących adresem sieci i *host-id* (host-idetifier), będący adresem host komputera w tej sieci. Na rys. 1 pokazano formaty trzech klas (rodzajów) adresów, stosowanych w adresacji sieci Internet, o różnej liczbie podsieci i host komputerów. Klasa A jest formatem do kodowania adresów najmniejszej liczby sieci (7 bitów adresowych) zapisywanych za pomocą identyfikatora sieci – net-id i największej liczbie host komputerów – host-id. Razem 24 bity adresowe. Klasa C – odwrotnie, dotyczy dużej liczby sieci i małej liczby *hostów* w sieci. Natomiast klasa B jest rozwiązaniem pośrednim.

Adres internetowy może być użyty do wskazania samej sieci lokalnej, wybranego host komputera lub grupy hostów. Jeśli host-id zawiera same zera (żaden z hostów) wskazywana jest droga przejścia, czyli adres węzła sterującego sieci. Same jedynki w pozycji hast-id zostały zarezerwowane dla adresu globalnego wszystkich host komputerów zainstalowanych w węźle. Jest to przypadek wykorzystywany dla procedur „rozgłaszania" (broadcasting) w sieci rozległej. Same zera na pozycji net-id oznaczają komunikację między host komputerami w ramach jednej sieci.



Rys. 1. Formaty klas adresowych dla Internet

Przejścia między sieciami określa net-id. Są one podstawą przełączeń międzysieciowych dla protokołów bezpołączeniowych. Protokoły bezpołączeniowe funkcjonują dzięki tablicom adresowym, rezydującym w węzłach sterujących lub bramach (gateway) węzła sieciowego, zakładanym w czasie transmisji rozgłoszeniowej. Identyfikatory sieciowe net-id są zapisywane w tablicach przełączeń (routingu) bram międzysieciowych. Pakiety rozgłoszeniowe mogą być wysyłane do jednego lub wielu odbiorców dla realizacji zadań, które umożliwiają zarejestrowanie adresów włączonych do sieci stacji i utworzenia tablic konfiguracji sieci. Identyfikatory adresują kanały przejścia informacji

przez łączniki (routery, mosty, itp.), od nadajnika do odbiornika pakietów danych.

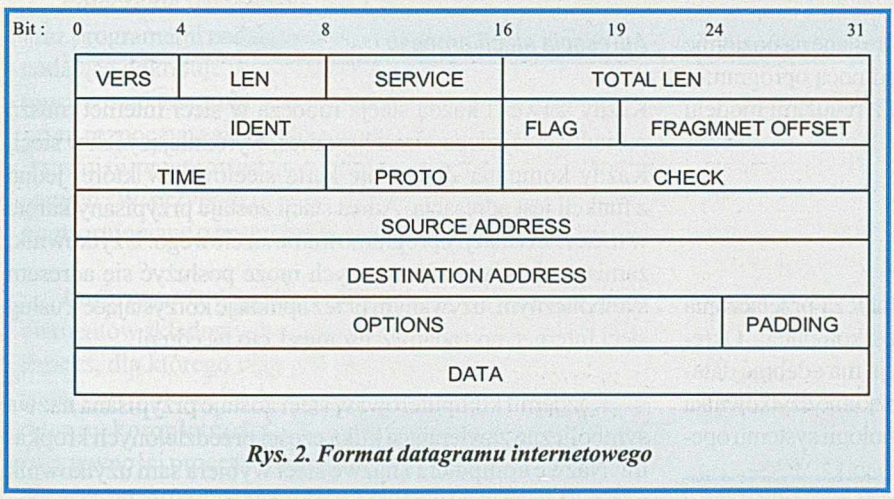
Komunikacja w sieci Internet jest określana 32-bitowym adresem binarnym zapisywanym w formie czterech liczb dziesiętnych oddzielonych kropkami, np. adres 10000001 00001011 0000011 00001110 jest zapisywany następującymi odpowiednikami dziesiętnymi 129.11.3.14.

IP – jest międzysieciowym (internetowym) protokołem bezpołączeniowym dla transportu danych w ramkach datagramowych. W ramach protokołu IP zdefiniowano format adresów sieciowych (opisany powyżej), format datagramu oraz zasady organizacji tablic adresowych przełączania dróg transmisji (routingu) datagramów.

Datagram – zawiera nagłówek, numer bloku oraz pole danych. Wędruje on przez kolejne routery sieci, od bramy do bramy, czy od bramy do docelowego host komputera. Choć w systemach transmisji danych z przełączaniem pakietów dopuszcza się przesył krótkich ciągów, do 128 oktetów, a nawet mniej, datagramowa transmisja takich małych zbiorów jest nieoptymalna. Choć standardy Internet nie limitują długości datagramów, stosuje się zalecenia rozwiązań dla mostów, polegające na fragmentacji danych w ramki o długościach do 576 oktetów. Bloki dłuższe poddawane są podziałowi na fragmenty typowe. Każdy fragment danych posiada nagłówek, tak jak każdy element datagramu.

Format datagramu IP

Na rysunku 2 pokazano format datagramu internetowego. Zawiera on sześć 32-bitowych pól nagłówka i pole danych zasadniczych.



Rys. 2. Format datagramu internetowego

- *VER* – jest czterobitowym kodem wersji datagramu pozwalającym potwierdzić zgodność wersji, w której nadajnik, odbiornik oraz bramy między nimi przesyłają ten sam format datagramu;
- *LEN* – jest czterobitowym kodem długości datagramu, która jest mierzona liczbą słów 32-bitowych;
- *SERVICE* – określa typ transmitowanych danych oraz zadanie, które mają wykonać;
- *TOTAL LEN* – definiuje całkowitą długość datagramu; nagłówka i bloku danych zasadniczych;
- *IDENT* – pole numeru fragmentu transmitowanych danych;
- *FLAGS* – pole znaczników informujących o tym, czy możliwa jest fragmentacja datagramu lub czy datagram został podzielony na fragmenty;
- *FRAGMENT OFFSET* – jest polem offsetu adresowego dla adresacji kolejnych fragmentów datagramu;
- *TIME* – jest polem oznaczającym dopuszczalny czas przebywania (w sekundach) datagramu w sieci Internet;
- *PROTO* – określa format i typ bloku danych, jest kodem protokołu transmisyjnego pola danych;
- ***CHECK*** – jest sumą kontrolną dla nagłówka datagramu;
- *SOURCE ADDRESS* – 32-bitowy adres źródła informacji;
- *DESTINATION ADDRESS* – 32-bitowy adres miejsca przeznaczenia;
- *OPTIONS* – jest polem używanym dla testowania sieci;
- *PADDING* – stanowi wypełnienie oktetami z samych zer, które uzupełniają nagłówek datagramu do pełnych słów 32-bitowych;
- *DATA* – jest polem zawierającym blok danych zasadniczych.

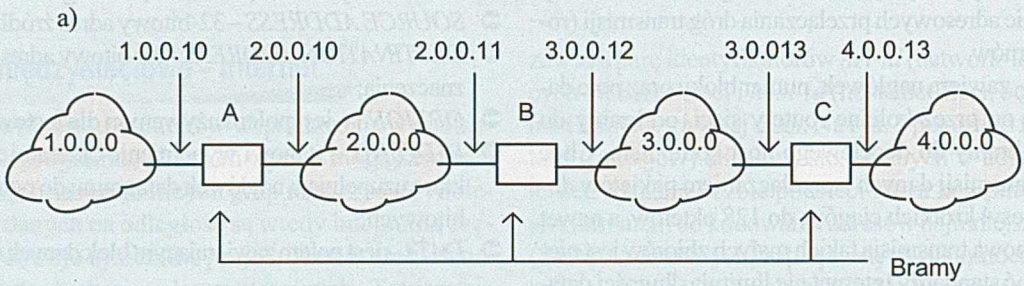
Tablice adresów przełączeń

Protokoły datagramowe transmisji bezpołączeniowej przekazują w linię transmisyjną ciąg pakietów informacji bez sprawdzania ich odbioru. Nie jest sprawdzane istnienie odbiorcy oraz nie są analizowane inne parametry przesyłu danych. W sieciach z protokołami bezpołączeniowymi zadaniem podstawowym staje się metoda rozpoznania adresu odbiornika (Address Resolution Problem) wskazywanego zawartością pakietu. Podstawowe algorytmy rozpoznawania kolejnych segmentów adresu przeznaczenia (routing) korzystają z tablic adresów zakładanych w węzłach sterujących pracą sieci, to znaczy w routerach, serwerach lokalnych i host komputerach, spełniających rolę serwerów globalnych. W rozbudowanych sieciach komputerowych liczba możliwych dróg przejścia jest tak duża, że wykluczona staje się możliwość pełnego ich zarejestrowania, w celu wskazania drogi do wybranego host komputera. Zmniejszenie wymiaru tablic adresowych umożliwia ograniczenie zapisu do identyfikatorów adresowych sieci. Na rysunku 3a pokazano oznaczenia adresów internetowych, a na rys.3b przykład zapisu tablicy adresowej zarejestrowanej w bramie (gateway) oznaczonej literą B.

Powyższy przykład pokazuje opis dróg komunikacji bramy B ze wszystkimi sieciami. Adresację hosta wewnątrz wybranej sieci zapewnia adres zapisany w datagramie przekazywanych danych. Sieci o identyfikatorach 2.0.0.0 i 3.0.0.0 mają bezpośrednie połączenie z B. Dalsze sieci są wskazywane numerem linii (hosta) dla najkrótszego połączenia bramy z tą siecią.

Internet jest ogólnoświatową siecią rozległą zawierającą setki tysięcy komputerów, specjalizowanych terminali i różnego typu sieci lokalnych. Wykorzystują one protokół IP, definiujący zasady adresacji podsieci i format datagramu. Pracujące

w sieci Internet sieci lokalne korzystają z własnych technologii komunikacyjnych, jak: Ethernet, FDDI, Token Ring i inne. Aktualnie największą grupę użytkowników Internetu tworzą sieci LAN komputerów osobistych, pracujących z systemami operacyjnymi MS Windows i oprogramowaniem komunikacyjnym z Internet. Sieci lokalne komunikują się z serwerami Internet pracującymi w większości przypadków pod systemem operacyjnym Unix. Użytkownik sieci Internet nie zauważa faktu niezgodności jego systemu operacyjnego z systemem serwera



b)

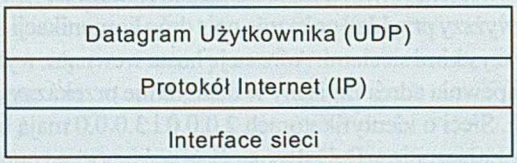
Adres sieci	Trasa do sieci
2.0.0.0	Dostęp bezpośredni
3.0.0.0	Dostęp bezpośredni
1.0.0.0	2.0.0.10
4.0.0.0	3.0.0.13

Rys. 3. Format adresu Internetowego
a) oznaczenia adresów
b) tablica adresów dla bramy B

głównego (Windows czy Unix). Zadania określane na poziomie terminala są przenoszone do serwera za pomocą oprogramowania i sprzętu transmisyjnego zgodnego z regułami modelu warstwowego OSI-7.

Format datagramu użytkownika – UDP

Operacje wykonywane na poziomie IP dotyczą przełączania datagramów z dokładnością do podsieci i host komputera. Określenie, który użytkownik lub która aplikacja ma odebrać datagram, wymaga dodatkowych określeń na poziomie użytkownika - UDP (User Datagram Protocol). W terminologii systemu ope-



Rys. 4. Schemat warstwowy protokołów w sieci Internet

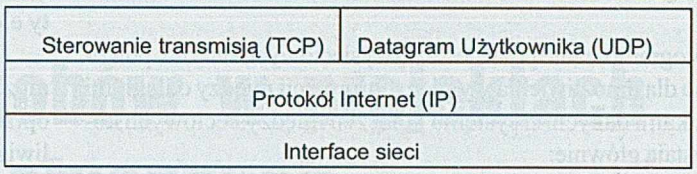
racyjnego winien zostać określony adres procesu użytkownika, natomiast w miejsce adresu procesów wprowadzono ponumerowane porty, do których kierowane są transmitowane wiadomości. Tak więc protokół UDP dołącza dodatkowe dwa bajty adresowe: portu przeznaczenia oraz portu źródłowego. Adres portu źródłowego jest podawany (opcjonalnie) w przypadkach, gdy oczekiwana jest informacja kwitująca poprawność transmisji. UDP jest protokołem zlokalizowanym nad protokołem internetowym (IP), zgodnie ze schematem warstwowym pokazanym na rys.4. Umieszczenie datagramu użytkownika (UDP) ponad IP oznacza, że datagram jest przesyłany według zasad opisanych dla IP.

Adresacja stacji/procesu

Każdy serwer i każda stacja robocza w sieci Internet muszą posiadać swój niepowtarzalny adres, wyróżniający ich w sieci. Każdy komputer dysponuje kartą sieciową, w której jedną z funkcji jest adresacja. Adres stacji zostaje przypisany karcie w trakcie instalacji oprogramowania sieciowego. Użytkownik, zamiast adresów numerycznych może posłużyć się adresem symbolicznym, używanym przez aplikacje korzystające z usług sieci Internet, np.: adam@.usctoux1.cto.us.edu.pl.

Każdemu komputerowi w sieci zostaje przypisana nazwa symboliczna zawierająca kilka części przedzielonych kropkami. Nazwę komputera i nazwę sieci wybiera sam użytkownik w czasie konfigurowania swojej LAN. Systemem nazw (domenami) zarządzają specjalne serwery w sieci – DNS (Domain Name System). Wszystkie serwery DNS sieci Internet są zorganizowane w strukturę hierarchiczną. Ich podstawowym zadaniem jest wyszukiwanie adresów poszczególnych sieci i komputerów w sieci, ale równie ważna jest konwersja nazwy przypisanej konkretnemu stanowisku sieciowemu na numer kodowy używany w protokole IP i odwrotnie – w sieci przeznaczenia włączony zostaje protokół ARP (Address Resolution Protokol), który wybiera odpowiedni adapter stanowiska sieciowego. Adres UDP jest zawsze 48-bitowy. Jest on przy-

pisany każdemu z adapterów (kart) sieciowych. Stację sieci Internet można zaadresować za pomocą jej nazwy lub numeru używanego przez protokół IP.



Rys. 5. Schemat warstwowy protokołów IP, UDP i TCP w sieci Internet

Protokół negocjacyjny TCP

Protokoły omówione poprzednio organizują transport pakietów danych, które są składową wiadomości; elementem datagramów. Ponieważ transmisja danych nie odbywa się bez zakłóceń, pakiety mogą zostać uszkodzone oddziaływaniem zakłóceń zewnętrznych. W wyniku awarii układów transmisyjnych mogą również zostać zagubione niektóre bloki transmitowanych danych, w rezultacie wyboru niedrożnych dróg transmisji. W systemach z dynamicznym (datagramowym) przełączaniem pakietów odbiór może nastąpić w nieuporządkowanej kolejności, mogą wystąpić opóźnienia w transmisji lub odebrane zostają duplikaty bloków danych. Wysoko zaawansowane aplikacje dokonują przesyłu bardzo dużych ciągów danych, dla których musi zostać zagwarantowana duża niezawodność transmisji.

Niezawodną transmisję ciągu danych gwarantują wyszczególnione niżej funkcje oprogramowania sterującego – TCP (Transport Control Protocol).

Orientacja strumienia danych – polegająca na badaniu kompletności odebranego ciągu danych, to znaczy, czy odebrane zostały wszystkie oktety nadane na wejściu linii transmisyjnej.

Połączenie wirtualne – ma miejsce między aplikacjami oraz programami nadającymi i odbierającymi dane. Maszyna nadająca dokonuje wywołania procesu odbierającego, który musi zostać zaaprobowany przez maszynę odbierającą, po czym rozpoczyna się przekazywanie danych. Na zakończenie transmisji przeprowadzona zostaje weryfikacja odebranych danych. W przypadku błędu transmisji obie strony zostają poinformowane o potrzebie powtórzenia operacji przesyłu.

Buforowania kanałów – które umożliwia gromadzenie elementów składowych ciągu danych wysyłanych pakietami. Proces, dla którego ciąg jest przeznaczony, zostaje poinformowany o gotowości danych po ich zweryfikowaniu i stwierdzeniu kompletności. W celu uzyskania dostatecznej efektywności procesu transmisji, konieczna jest koncentracja danych po stronie nadawania, zgodnie z formatem datagramów obowiązujących w systemach transmisji sieci rozległych.

Formatowanie ciągu danych – dopasowuje przesyłane pliki do formatów obowiązujących w sieciach internetowych.

Transmisja obustronna – oznaczająca możliwość bieżącego przesyłu danych w obu kierunkach linii (full duplex).

TCP (Transport Control Protocol) – spełnia rolę protokołu negocjującego warunki selekcji trasy – kanału wirtualnego (logicznego) dla najlepszej drogi transmisji. Odpowiada on za utrzymanie ustalonych warunków transmisji na czas połączenia oraz gwarantuje dostarczenie pakietów do odbiorcy i sprawdza poprawność przesłania danych. Podobnie jak dla warstwy UDP można zdefiniować lokację warstwową dla protokołów sterujących TCP. Relacje poszczególnych serwisów internetowych zilustrowano na rys. 5.

Modemowe podłączenie do internet

Szczególnym przypadkiem sieci Internet jest terminal modemowy. Jest to zazwyczaj komputer osobisty łączący się z serwerem Internet za pomocą modemów telefonicznych. Takie komputery nie posiadają standardowej karty sieciowej LAN, nie są stałymi abonentami sieci i nie są bezpośrednio adresowane. Transmisja danych za pomocą kart modemowych i odpowiedniego oprogramowania odbywa się zwykle z użyciem standardowych, publicznych, komutowanych linii telefonicznych. Komputer staje się nieinteligentnym terminalem do przesyłania plików. Korzystanie w sieci Internet z takiego terminala jest możliwe po założeniu w serwerze konta terminala i jego wpisaniu do listy użytkowników. Lista użytkowników sieci jest chroniona hasłem. Istotną cechą rozwiązania z terminalem nieadresowanym jest przetwarzanie głównych zadań transmisyjnych użytkownika nie na stacji roboczej, a na serwerze.

Można również spotkać rozwiązania z adresacją terminali, które wykorzystują oprogramowanie komunikacyjne z protokołami SLIP (Serial Line Internet Protocol) lub PPP (Point-to-Point). Komputer jest w dalszym ciągu jednym z terminali serwera, ale przetwarzanie odbywa się na końcówce użytkownika, a nie na odległym serwerze, do którego jest podłączony nasz terminal. Dostęp do plików przechowywanych na odległych serwerach realizuje program FTP (File Transport Protocol), przenoszący wskazany plik na terminal użytkownika sieci.

Anonimowy dostęp do danych (Anonymous File Transfer Protocol, FTP) – reprezentuje zasady dostępu do danych nie chronionych żadnymi hasłami. Systemy komputerowe podłączone do sieci Internet udostępniają te dane na konto anonymous – użytkownika anonimowego. Dostęp do takich danych następuje po uruchomieniu programu FTP ze wskazaniem komputera pożądanego i zalogowaniu się (włączeniu się) do sieci jako użytkownik anonimowy (anonymous). W oknie hasło można podać własny adres pocztowy, następnie podajemy hasło dostępu – quest i można pobrać udostępnione dane za pomocą programów narzędziowych dla przesyłania plików.

Poczta elektroniczna

Oprogramowanie połączeń międzysieciowych Internet powołano dla umożliwienia wymiany informacji między odległymi bankami danych. Z systemu połączeń międzysieciowych korzystają głównie:

- poczta elektroniczna (E-mail): do przesyłu pilnej korespondencji w formie krótkich plików tekstowych), przesyłu dokumentów zawierających długie pliki tekstowe i graficzne, prowadzenia dyskusji w grupach, itp.;
- biblioteki oprogramowania i bazy danych;
- użytkownicy komputerów dużych mocy obliczeniowych (Main Frame) przekazujących wsad do komputera za pośrednictwem odległego terminala.

Niewątpliwie najczęściej wykorzystywaną formą komunikacji między użytkownikami sieci rozległych jest poczta elektroniczna (Electronic Mail; E-mail). Jest to metoda przekazywania komunikatów między stacjami sieci rozległych. Implementacja poczty elektronicznej zawiera:

- narzędzia do tworzenia korespondencji;
- metody adresowania stacji odbierających, wysyłania i odbierania wiadomości i odpowiadania na korespondencję;
- metody dołączania plików danych do komunikatu pocztowego aplikacji.

Poczta elektroniczna może zostać obsłużona za pomocą komputera o dowolnych parametrach i dowolnym oprogramowaniu operacyjnym. Przekazywanie plików tekstowych umożliwiają programy inicjujące tryb transmisji danych za pomocą odpowiednich kart (urządzeń) komunikacyjnych. Komputery podłączone do odległych końców sieci telekomunikacyjnej uruchamiają mechanizmy szeregowego przesyłu znaków ASCII. Po obu stronach sieci telekomu nikacyjnej zainstalowane jest to samo oprogramowanie i takie same urządzenia komunikacyjne.

Problem obsługi poczty elektronicznej staje się bardziej złożony, gdy wymiana informacji ma się odbywać między komputerami o różnych, wzajemnie niezgodnych, standardach systemowych (operacyjnych). Tego rodzaju niezgodności są eliminowane za pomocą specjalnych systemów pośredniczących:

- za pomocą bram (Gateway) międzysystemowych dla poczty elektronicznej;
- za pomocą uniwersalnych sieci szkieletowych (E-mail backbone) dla poczty elektronicznej.

E-mail szkielet – jest rozwiązaniem uniwersalnym dla wszystkich systemów. Oznacza on konieczność dopasowania każdego ze stanowisk poczty do wspólnego protokołu transmisyjnego. Każde stanowisko sieci musi być wyposażone w bramę (Gateway) przenoszącą format danych terminala na uniwersalny format szkieletu. Ta uniwersalna magistrala teletransmisyjna nazywana jest szkieletem poczty elektronicznej lub siecią szkieletową. Uniwersalność szkieletu oznacza możliwość unifikacji protokołów dla poczty elektronicznej.

Do grupy uniwersalnych protokołów poczty elektronicznej można zaliczyć standard międzynarodowy X.400, opracowany przez organizację CCITT. Protokół X.400 umożliwia każdemu z użytkowników sieci dostosować swoje urządzenia pośredniczące (interface) do jednolitego standardu centralnego zwanego szkieletem. Urządzenia transmisyjne szkieletu w standardzie X.400 realizują trzy podstawowe funkcje:

- odczytu wiadomości w formacie urządzenia nadającego (Message User Agent);
- określenia adresu odbiorcy przeznaczenia (Message Router);
- przesłania w odpowiednim formacie wiadomości do odbiorcy przesyłki pocztowej (Message Transfer Agent).

Każde ze współczesnych oprogramowań systemowych, zaprojektowanych do pracy w sieci komputerowej zostało wyposażone w oprogramowanie do transmisji wiadomości (messaging application). Dostosowanie poczty elektronicznej systemu operacyjnego do standardu X.400 realizują dodatkowe programy aplikacyjne (Application Program Interface – API) dostosowujące formaty transmisyjne standardowych kart (Ethernet, Token-Ring lub inne) do standardu X.400; formatu szkieletu.

Serwer poczty – zajmuje się obsługą węzła poczty elektronicznej a w szczególności:

- zapisywanie w pamięci węzła otrzymanych plików;
- zakładaniem i utrzymywaniem kont użytkowników;
- ochroną tajemnicy korespondencji;
- tworzeniem grup użytkowników i list indywidualnych adresów;
- przekazywaniem przesyłek pocztowych.

Literatura

[1] Comer D.E. Internetworking with TCP/IP. Principles, protocols, architecture. Prentice-Hall International, Inc. USA 1988,
[2] Goldman J.E. Applied Data Communications. A bussiness-oriented approach. J.Willey & Sons, Inc. USA, 1995,
[3] Madron T.W. Local Area Networks. New Technologies, Emerging Standards. J. Willey & Sons, Inc. USA, 1994,
[4] Mirchandani S., Khana R. FDDI Technology and Applications, J. Willey & Sons, Inc. USA, 1993.

Prace realizowane w ramach badań statutowych BS202/IIS/97

prof. Jan Piecha jest pracownikiem Instytutu Informatyki Stosowanej na Uniwersytecie Śląskim
piecha@us.edu.pl.

Szacowanie zasobów w projektach informatycznych

Marek Miłosz

Ile to będzie kosztować? Jak długo potrwa? Kiedy skończycie? Co jest potrzebne do realizacji? ... Z takimi lub podobnymi pytaniami spotyka się kierownik projektu informatycznego na każdym jego etapie: od rozmów wstępnych do wdrożenia już gotowego systemu. Prawidłowość oszacowania zasobów niezbędnych do realizacji projektu jest sztuką, od której opanowania zależy sukces przedsięwzięcia. Jak zatem precyzyjnie określać wymagane zasoby: czas, koszty, liczebność zespołu, wartość materiałów, ...

Projekty informatyczne są z wielu powodów trudne w planowaniu i realizacji. Oprogramowanie bowiem charakteryzuje się całą listą cech, które odróżniają je od dowolnie innego, materialnego produktu. Jego abstrakcyjność i duża złożoność powoduje znaczne trudności na wszystkich etapach realizacji projektu informatycznego. Projektom informatycznym brakuje efektu skali, tak charakterystycznego dla produktów materialnych. Efekt skali powoduje, że jednostkowe koszty produkcji maleją przy wzroście jej rozmiarów. W przedsięwzięciach informatycznych jest na ogół odwrotnie. Przy wzroście złożoności systemu koszt jego wytworzenia wzrasta nieproporcjonalnie. Efekt ten jest związany z lawinowym wzrostem stopnia komplikacji systemu informatycznego (mierzonego np. liczbą związków pomiędzy jego poszczególnymi elementami systemu) przy wzroście jego wielkości.

W projektach informatycznych mamy także do czynienia z paradoksalną, zdawałoby się, sytuacją. Częstokroć, rozpoczynając projekt informatyczny, nie wiemy dokładnie, w jakiej technologii i za pomocą jakich narzędzi będzie on zrealizowany. Dzieje się tak w wyniku naturalnego dążenia twórców systemu do unowocześnienia rozwiązań oddawanych do użytkowania. Zwykle od rozpoczęcia projektu do jego ukończenia rzadko upływa mniej niż rok czy dwa. Jest to okres, w którym zmieniają się nie tylko wersje produktów informatycznych, ale nawet całe technologie. Z tego powodu w rozpoczynanych dwa lata temu projektach rzadko znajdziemy takie terminy, jak: Intranet, Java czy też praca grupowa.

Projekty informatyczne są zwykle źle uwarunkowane. Nie stoją za nimi setki i tysiące lat doświadczeń ludzkości (jak np. w budownictwie czy medycynie) lub też duże, zbiorowe doświadczenie i dobrze sprecyzowane wyniki (jak np. przy programowaniu gier w systemie Windows). Projekt informatyczny oznacza zawsze nową sytuację, nieznane warunki realizacji, a także ograniczony zakres i przydatność doświadczeń.

Tymczasem zwykle po wstępnym zdefiniowaniu potrzeb, a co za tym idzie zakresu projektu, zwykle padają pytania: kiedy? za ile? za pomocą czego? Odpowiedzi na te i inne tego typu pytania wymagają dokładnego oszacowania różnorodnych zasobów niezbędnych do realizacji projektu.

Tendencje rozwojowe systemów informatycznych wykazują jednoznacznie wzrost udziału kosztów wytworzenia oprogramowania (a więc analizy, projektowania, kodowania, testowania, wdrożenia, utrzymania i rozwoju) w kosztach ogólnych systemów. Czynnik, jakim jest koszt, który w dużym stopniu można przewidzieć, staje się coraz mniej ważny w wielu dziedzinach zastosowań informatyki. Proces ten utrudnia szacowanie pracochłonności, a zatem i kosztów projektów informatycznych.

Szacowanie i jego zakres w projekcie informatycznym

W projektach szacuje się te nieznane wielkości, których nie można wyliczyć, wyznaczyć lub wartości, które nie wynikają w sposób jednoznaczny z zewnętrznych uwarunkowań. Wpływa na nie wiele różnych zmiennych elementów natury ludzkiej, technicznej i zewnętrznej - z otoczenia projektu informatycznego. Rezultatem procesu oszacowania jest więc probabilistyczna wielkość - szacunek, estymator. Estymacja uogólnia ciąg doświadczeń pojedynczej osoby, zbiór jednostkowych doświadczeń wielu ludzi lub też oba elementy łącznie. Estymację otrzymuje się przy pewnych założeniach, mniej lub bardziej oczywistych.

Oszacowaniu w projekcie informatycznym podlega wiele rzeczy- począwszy od najbardziej naturalnych, takich jak: pracochłonność, wydajność pracownika, wielkość zatrudnienia w każdej fazie projektu, koszty w różnych przekrojach i czasie, zasoby materialne potrzebne do realizacji projektu (urządzenia i ich obciążenie, materiały, itp.), aż po mniej oczywiste: straty czasu na działania nieprodukcyjne, kwalifikacje pracowników, jakość dostaw i obsługi, zagrożenia realizacji projektu. Nawet takie elementy, jak założenia, według których realizowany jest projekt, są w wielu przypadkach jedynie szacunkiem. Szacuje się bowiem czas pracy w firmie (nie zawsze jest to przepisowe osiem godzin), liczbę podróży służbowych do klienta, czas dojazdu, itp.

Można zaryzykować stwierdzenie, że szacowaniu podlegają wszystkie elementy, których koszty mogą w istotny sposób wpływać na ostateczny koszt realizacji projektu. Dokładność szacunków zależy od wielkości tych kosztów - tab. 1. Stopień ważności i rola oszacowania rośnie wraz ze wzrostem zakresu, wielkości i kosztów projektu informatycznego. Podobnie jak konsekwencje błędu w szacunkach.

Estymacja różnych wielkości w projekcie nie jest stabilna, bowiem w trakcie realizacji projektu wiele szacunków podlega ponownym ocenom. Związane jest to z następującymi procesami:

- stopniowym wzrostem dokładności ocen w miarę realizacji projektu (wzrost wiedzy o projekcie);
- oddziaływaniem różnorodnych czynników (wewnętrznych i zewnętrznych) na proces realizacji projektu, prowadzącym do konieczności modyfikacji założenia ocen;
- modyfikacją warunków, zakresu i innych cech projektu.

Klasy dokładności estymacji *Tabela 1*

Klasa	Typ kosztów, związanych z ocenianym elementem	Dokładność
I	Główny, określające koszt końcowy	±5%
II	Wysoki, znacząco wpływające na koszt końcowy	±10-15%
III	Poboczny, z możliwymi elementami znaczącymi	±15-20%
IV	Poboczny o umiarkowanym wpływie	±20-25%
V	Widoczny	±25-35%
VI	Niezauważalne, bardzo niewiele wpływające na koszt końcowy	>±35%

Źródło: [10]

Problemy i zagrożenia związane z szacowaniem projektów

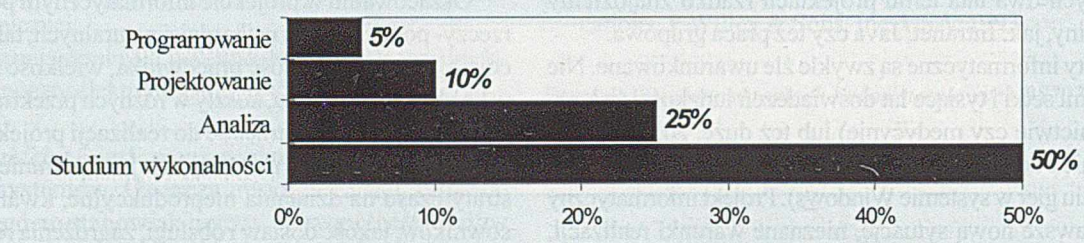
Yourdon [15] wskazuje wiele zagrożeń związanych z szacowaniem w projektach informatycznych. Do podstawowych z nich należą:

- różnica pomiędzy szacowaniem (realizm) a negocjacjami (kompromis);
- konieczność uwzględnienia nie średnich, ale realnych zdolności zespołu realizującego projekt (w tym, bardzo często, i zdolności uczenia się - „krzywa uczenia");
- duża niepewność samooceny twórców systemów informatycznych;
- brak dokładnych danych historycznych;
- nacisk kierownictwa na zbyt szczegółowe i przedwczesne oszacowania;
- optymistyczne założenia dotyczące pracy w nadgodzinach, „szturmowania terminów";
- trudności w ustanowieniu miary produktywności analityków, projektantów i programistów.

W trakcie negocjacji często uzyskuje się kompromis na nierealnych warunkach, które potem nie mogą być dotrzymane. W związku z tym w procesie negocjowania należy na bieżąco dokonywać zmian w już ocenionych parametrach. Nie można, w imię pozyskania kontraktu, zmieniać w dowolny sposób wcześniej opracowanych szacunków zasobów niezbędnych do realizacji projektu. Takie podejście wcześniej lub później obraca się przeciwko realizatorom projektu, powodując wydłużenie czasu jego realizacji czy też wzrost kosztów.

Nacisk kierownictwa i użytkownika na bardzo szczegółowe oceny na początkowych etapach prac nad systemem informatycznym jest zjawiskiem powszechnym i naturalnym. Wiąże się on z chęcią zaplanowania globalnych nakładów i zasobów potrzebnych do budowy systemu. Niestety, często te szacunki są potem wpisane do umów i, w konsekwencji, bardzo ściśle przestrzegane. Tymczasem na wstępnych etapach projektu błąd szacowania jest z natury rzeczy bardzo duży i maleje w miarę postępu prac (pod warunkiem dokonywania kolejnych estymacji) - rys. 1.

Problem pomiaru produktywności występuje nie tylko w dziedzinie inżynierii oprogramowania. Słabe możliwości mie-



Źródło: opracowanie własne na podstawie [15]

Rys. 1. Dokładność szacowania na poszczególnych etapach projektu informatycznego

rzenia rezultatów projektu i duża różnorodność wykorzystywanych narzędzi programistycznych potęgują te problemy. Wykorzystywany, wydawałoby się, naturalny miernik wielkości oprogramowania, linie kodu źródłowego (ang. *LOC - Lines Of Code*) lub tzw. punkty funkcyjne (ang. *FPs - Function Points*), wprowadzone przez Allana J. Albrechta w 1979 roku [2], nie stanowią z wielu względów miary na tyle uniwersalnej i oczywistej, by mogły być w prosty sposób wykorzystane przy estymacji parametrów projektów informatycznych.

Innym niemniej ważnym problemem szacowania pracochłonności projektów informatycznych jest bardzo złożona zależność między pracochłonnością mierzoną np. w osobodniach a realnym, kalendarzowym czasem realizacji projektu. Nie jest bowiem oczywiste, że czas realizacji projektu zależy wprost proporcjonalnie od pracochłonności, a odwrotnie proporcjonalnie od liczebności zespołu.

Metody szacowania projektów

Szacowanie zasobów w projektach informatycznych jest kombinacją dobrych danych historycznych (doświadczeń) i metod ich wykorzystania w nowych przypadkach. Metody są różne i zwykle wykorzystywane łącznie, tj. do oszacowania zasobów projektu wykorzystuje się równocześnie dwie metody lub więcej. Można wyróżnić następujące możliwe sposoby szacowania zasobów projektów informatycznych:

- opóźnianie oszacowań;
- analogia;
- metody dekompozycyjne;
- ekstrapolacja przy pomocy różnych empirycznych modeli parametrycznych.

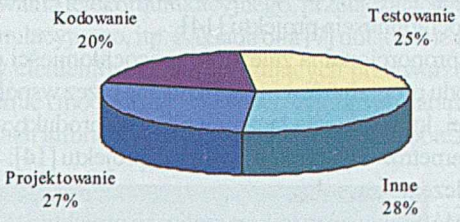
Opóźnianie procesu oszacowania pracochłonności i przenoszenie go na późniejsze etapy projektu ma swoje uzasadnienie w prostej zasadzie - im później oceniasz,tym więcej wiesz. Skuteczność tego podejścia odwzorowuje rys. 1. W praktyce jednak, z wielu oczywistych względów, nie udaje się skutecznie stosować tej „metody".

Szacowanie przez analogię bardzo często jest porównywane do „metody sufitu". Wymaga ono posiadania dużego doświadczenia w realizacji podobnych projektów. Można w tej metodzie opierać się także na doświadczeniach innych projektów informatycznych. Doświadczenia te, ujęte w wykresy i tabele pracochłonności poszczególnych etapów projektów (np. rys. 2),umożliwiają, po oszacowaniu nakładów na jeden etap, wyznaczenie nakładów na cały projekt.

Metody dekompozycyjne polegają na połączeniu metody *top-down* i *bottom-up*. Ich idea zakłada podział projektu informatycznego na wiele szczegółowych zadań (dekompozycja), oszacowanie pracochłonności każdego z nich i wyznaczenie na tej podstawie (synteza) pracochłonności całego projektu. Realizują one następujący schemat: **dekompozycja → estymacja → synteza**.

Metoda dekompozycji działań opiera swoją poprawność na dwóch przesłankach:

- statystycznym wyrównywaniu błędów (działanie tzw. centralnego twierdzenia granicznego);



Źródło: [13]

Rys. 2. Pracochłonność poszczególnych etapów projektów informatycznych

- znacznie większej potencjalnej dokładności estymacji małych zadań w porównaniu z dużymi.

W związku z tym, im więcej zadań cząstkowych zostało wyróżnionych w projekcie, tym dokładniejsza będzie estymacja całości. Oczywiście nie należy przesadzać w procesie podziału. W projektach informatycznych Yourdon [15] zaleca stosowanie dekompozycji do zadań, których pracochłonność mierzy się osobotygodniami.

Metody dekompozycyjne wspierane są całym szeregiem technik: strukturyzacją prac, harmonogramowaniem (wykresy Gantta) i technikami sieciowymi (CPM, PERT, GAN). Większość technik jest wspomagana przez odpowiednie narzędzia komputerowe, które wymagają danych wejściowych: struktury prac (i ich szeregowania) i estymacji parametrów poszczególnych zadań. Wspomagają więc pracę tylko instrumentalnie. Problem szacowania przenoszony jest na niższy poziom, co wymaga dość szczegółowych informacji na temat projektu, jego produktu i struktury. Zwykle te informacje nie są dostępne na początkowych etapach rozmów nad projektem.

Metody ekstrapolacyjne polegają na budowie empirycznych modeli, pozwalających oszacować wielkości zasobów w projektach informatycznych na podstawie atrybutów produktywności. Modele są opracowywane na podstawie danych empirycznych. Uogólniają więc doświadczenia zbierane w trakcie realizacji wielu projektów.

Do najbardziej znanych modeli parametrycznych należą:

- model „linii kodu" firmy IBM;
- metoda punktów funkcyjnych (ang. *FPs - Function Points*) Allana Albrechta [2] i jej liczne modyfikacje;
- model COCOMO (ang. *COnstructive COst MOdel*) Barrego Boehma [5].

Modele „linii kodu"

Powstałe w latach 70. modele „linii kodu" są do chwili obecnej udoskonalane - głównie przez duże firmy programistyczne, firmy konsultingowe, instytuty naukowo-badawcze i uczelnie wyższe. Wykorzystują one najbardziej naturalną miarę oprogramowania - linie kodu programu. Ich ilość jest bowiem niewątpliwie dobrym wyznacznikiem stopnia złożoności systemu informatycznego.

Stosunkowo prosty i dość znany model „linii kodu", opracowany przez pracowników firmy IBM, wiąże tysiące linii kodu

programu (ang. *KLOC - Kilo Line of Code*) z miarą produktywności - pracochłonnością projektu [14].

Wprost proporcjonalna zależność pracochłonności od tysiąca linii kodu programu jest modyfikowana przez współczynnik trudności, który określa łączny wpływ na produktywność różnych parametrów i warunków realizacji projektu [14]:

- doświadczenie zespołu;
- zmienność wymagań w trakcie realizacji projektu;
- ograniczenia sprzętowe;
- poziom wymagań realizacji zadań;
- ograniczenia zewnętrzne.

Analogiczne modele opracowali inni autorzy [3].

Model „linii kodu" jest bardzo prostym modelem. Jego wykorzystanie bardzo ogranicza fakt, że do wyznaczenia pracochłonności projektu należy wcześniej oszacować ilość linii - KLOC. Nie jest to możliwe na wstępnych etapach projektowania systemów informatycznych. Ponadto zależności metody szybko się dezaktualizują w miarę rozwoju języków programowania, narzędzi wspomagających opracowanie systemów, itd. Formuły wyznaczające pracochłonność zależą bardzo od „jakości" (poziomu) języka programowania. Jeden KLOC w asemblerze znaczy całkiem co innego niż jeden KLOC w Fortranie, Pascalu czy też języku 4GL.

W metodzie „linii kodu" poważny problem stanowi samo pojęcie linii kodu. Czy komentarz w kodzie programu jest potrzebny i zliczany czy też nie? Linia linii nie jest równa. Co z poleceniami zajmującymi szereg linii? Czy deklaracje są liniami kodu?

Niska użyteczność modeli „linii kodu" wynika także z innego faktu. Mianowicie zużycie zasobów (w tym i pracochłonność) na kodowanie i testowanie nie przekracza na ogół 40-50% całkowitego zużycia w całym projekcie. Zwiększenie efektywności prac w wyniku zastosowania np. dwukrotnie „wydajniejszego" języka oprogramowania nie doprowadza do dwukrotnego zmniejszenia zasobochłonności projektu; właśnie ze względu na to, że złożoność i w konsekwencji koszt prac analityczno-projektowych mało zależą od języka programowania. Zależą one przede wszystkim od tego, co i w jaki sposób ma realizować aplikacja.

Metoda punktów funkcyjnych

Metoda punktów funkcyjnych (metoda FPs) została opracowana przez długoletniego pracownika firmy IBM- Allana J. Albrechta- w 1979 roku oraz modyfikowana przez niego w 1983 r. i rozwijana przez International Function Point Users Group (IFPUG) w latach: 1986, 1988, 1990 i 1994 [1].

Metoda punktów funkcyjnych wydziela parametry przetwarzania w systemie informatycznym jako atrybuty produktywności dla nieistniejącego systemu. Szacuje wpływ każdego z nich na produktywność i nakłada na tę ocenę szacunek wpływu warunków realizacji projektu informatycznego. Wyznaczone w ten sposób punkty funkcyjne są traktowane jako miara produktywności, a więc i złożoności, i- co za tym idzie- pracochłonności projektu informatycznego. Należy podkreślić, że projekt jest oceniany na wstępnych etapach z uwzględnieniem warunków jego późniejszej realizacji. Metoda FPs jest zatem bardzo cenna z punktu widzenia szacowania pracochłonności projektów. Jej idea może zostać w prosty sposób rozszerzona na szacowanie innych, niż pracochłonność, zasobów.

Metoda FPs umożliwia oszacowanie złożoności projektu informatycznego z uwzględnieniem dwóch różnych grup czynników:

- elementów przetwarzania informacji w systemie (w aspekcie funkcjonalnym i informacyjnym). Elementy te, to: wejścia i wyjścia z systemu, zbiory danych wewnętrznych i zewnętrznych i zapytania do bazy danych. Łączy się je w grupy o oszacowanych rozdzielnie wagach i sumuje z uwzględnieniem wag ich liczności. Ta ważona suma daje tzw. nie skorygowane punkty funkcyjne (ang. *UFP - Unadjusted Function Points*);
- całościowej oceny stopnia złożoności systemu i warunków jego realizacji w postaci różnorodnych czynników korygujących składających się na kompleksowy współczynnik korygujący (ang. *VAF - Value Adjustment Factor*). Czynniki korygujące określa się dla systemu jako całości, bez wnikania w jego wewnętrzną strukturę.

Nie skorygowane punkty funkcyjne wyznacza się jako ważoną sumę liczby elementów przetwarzania, występujących w systemie informatycznym, w pięciu grupach z uwzględnieniem wag o trzech poziomach wartości oceniających poziom złożoności elementu [8].

Do elementów przetwarzania A. Albrecht zaliczył:

- wejścia użytkownika;
- wyjścia użytkownika;
- zbiory danych wewnętrzne;
- zbiory danych zewnętrzne;
- zapytania zewnętrzne.

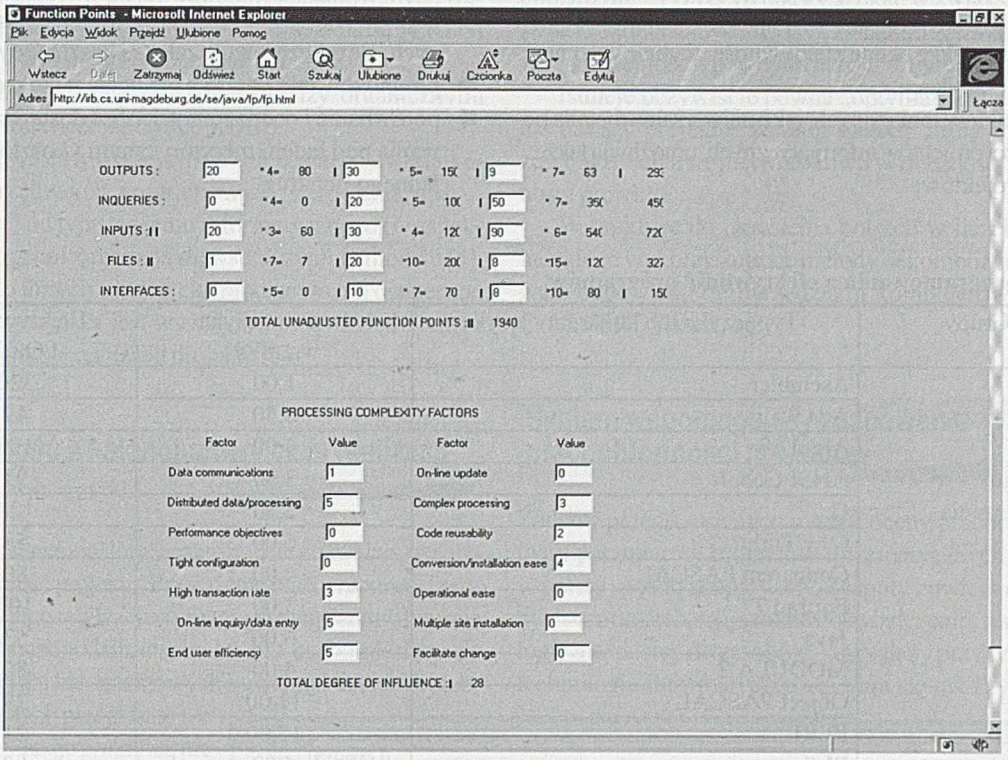
Ich wpływ na wielkość projektu informatycznego różni się znacznie: od 2 punktów do 15.

Wyznaczone w powyższy sposób nie skorygowane punkty funkcyjne podlegają modyfikacjom, uwzględniającym wpływ innych parametrów projektu.

Metoda punktów funkcyjnych wyróżnia następujące *czynniki korygujące*:

- występowanie urządzeń komunikacyjnych;
- rozproszenie przetwarzania;
- czas oczekiwania na odpowiedź systemu;
- stopień obciążenia systemu;
- częstotliwość wykonywania transakcji;
- wprowadzanie danych w trybie bezpośrednim;
- wydajność użytkownika końcowego;
- aktualizacja danych w trybie bezpośrednim;
- złożoność przetwarzania danych;
- możliwość ponownego użycia programów w innych zastosowaniach;
- łatwość instalacji;
- łatwość obsługi systemu;
- rozproszenie terytorialne;
- łatwość wprowadzania zmian - pielęgnowania systemu.

Wpływ każdego z powyższych czternastu czynników może być różny w danym projekcie. Jest on szacowany w skali sześciostopniowej: 0, 1, 2, 3, 4, 5. Im większy, bardziej znaczący wpływ danego czynnika, tym wyższa jego ocena punktowa. Łączny wpływ czynników korygujących uwzględnia ***komplek-*** gramowania telekomunikacyjnego, przełączającego, sterującego, symulacyjnego, itp. Zakłada ona [7] dołączenie algorytmów do listy elementów charakteryzujących przetwarzanie danych w systemie i modyfikację wag poszczególnych parametrów. Przy systemach informatycznych zarządzania (SIZ) obie metody dają



Źródło: http://irb.cs.uni-magdeburg.de/se/java/fp/fp.html

Rys. 3. Obliczenie punktów funkcyjnych przy pomocy apletów Javy

sowy współczynnik korygujący, który jest prostą sumą oszacowań wpływu każdego czynnika.

Punkty funkcyjne pierwotnie służyły do wyznaczania pracochłonności projektów informatycznych według empirycznie uzyskanych zależności [8].

Metoda punktów funkcyjnych ma zastosowanie do dzisiaj. W Internecie można znaleźć wiele adresów poświęconych temu problemowi. Dostępne są programy wspomagające prace obliczeniowe a nawet strony, które przy pomocy apletów Javy realizują obliczenia według formuł A.Albrechta: http://irb.cs.uni-magdeburg.de/se/java/fp/fp.html (rys. 3).

Na bazie idei metody punktów funkcyjnych opracowywane są inne. Jedną z nich jest metoda **punktów charakterystycznych** (ang. *FP - Feature Points*). Metoda ta powstała dla systemów operacyjnych, systemów czasu rzeczywistego, oprapraktycznie te same rezultaty. Wynika to ze znikomego wpływu złożoności algorytmów na pracochłonność projektu SIZ, a w zasadzie z braku konieczności opracowywania takich algorytmów. Rozbieżności pojawiają się w przypadku systemów, w których algorytmika odgrywa dużą rolę (tab. 2).

Z innych metod pokrewnych FPs należy zwrócić uwagę na: MKII i 3D FPs. Ta ostatnia metoda zakłada trójwymiarowy przekrój przez projekt informatyczny: dane, funkcje i sterowanie [7].

Z metodą punktów funkcyjnych związana jest także ocena produktywności języków programowania. Efektywność ta mierzona jest liczbą linii kodu potrzebnych do realizacji jednego punktu funkcyjnego (ang. *LOC/FP - Line of Code / Function Point*). Tabele przeliczników są tworzone dla różnorodnych systemów na podstawie analizy przebiegu zrealizowanych projektów systemów informatycznych (tab. 3). Przy pewnych

Współzależność punktów funkcyjnych i charakterystycznych *Tabela 2.*

Typ aplikacji	Punkty funkcyjne	Punkty charakterystyczne
Wsadowe SIZ	1	0,80
Interaktywne SIZ	1	1,00
Systemy telekomunikacyjne	1	1,20
Systemy kontroli	1	1,28
Wbudowane systemy czasu rzeczywistego	1	1,35
Automatyka przemysłowa	1	1,50

Źródło: [7]

systemach zatraca się jednak pojęcie linii kodu. Przykładem mogą służyć arkusze kalkulacyjne, w których linie kodu przyrównywane są do formuł, czy też generatory aplikacji, w których linie kodu to: obiekty i ich metody, instrukcje w 4GL, obsługa zdarzeń, wyzwalacze, itp.

Przeliczenie punktów funkcyjnych na linie kodu stanowi pomost pomiędzy metodą punktów funkcyjnych a metodą linii kodu. Umożliwia także ocenę efektywności wybranych narzędzi do tworzenia aplikacji.

Metoda punktów funkcyjnych, jako uniwersalna metoda pomiaru złożoności projektów informatycznych, umożliwia także:

- auditing projektów;
- wybór systemów informatycznych funkcjonujących w przedsiębiorstwie do reengineeringu;
- szacowanie liczby testów;
- ocenę jakości pracy i wydajności zespołów ludzkich;
- ocenę stopnia zmian wprowadzanych przez użytkownika na poszczególnych etapach budowy systemu informatycznego (wymaga to ciągłego przeliczania punktów funkcyjnych, ale odwzorowuje zakres i koszt modyfikacji);
- prognozowanie kosztów pielęgnacji i rozwoju systemów;
- porównanie i ocenę różnych ofert dostawców oprogramowania pod kątem merytorycznym i kosztowym (np. koszt jednego punktu).

Języki programowania a efektywność programów *Tabela 3*

Grupy	Typ języka lub konkretny język	Poziom języka wg. SPR	Efektywność, LOC/FP
Języki algorytmiczne i inne ogólnego przeznaczenia	Asembler	1,00	320
	Ada 95	6,50	49
	Algol 68	3,00	107
	ANSI Cobol	3,50	91
	C	2,50	128
	C++ Turbo	6,00	53
	Concurent PASCAL	4,00	80
	Fortran 77	3,00	107
	Java	6,00	53
	MODULA 2	4,00	80
	Object PASCAL	11,00	29
	PERL	15,00	21
	PL/I	4,00	80
	PASCAL Turbo	4,50	71
	Visual Basic 3	10,00	32
	Visual C++	9,50	34
Języki baz danych	Access	8,50	38
	ANSI SQL	25,00	13
	CLARION	5,50	58
	CA Clipper	17,00	19
	dBase III	8,00	40
	dBase IV	9,00	36
	DELPHI	11,00	29
	FOXPRO 2.5	9,50	34
	INFORMIX	8,00	40
	MAGIC	15,00	21
	ORACLE	8,00	40
	Oracle Developer/2000	14,00	23
	PROGRESS v. 4	9,00	36
	SYBASE	8,00	40
Języki symulacyjne	CSL	6,50	49
	DYNAMO-III	7,00	46
	GPSS	7,00	46
	SIMPLAN	9,00	36
Inne systemy	DOS - pliki wsadowe	2,50	128
	Excel 5	57,00	6
	HTML 3.0	22,00	15
	MathCAD	60,00	5
	Mosaic	50,00	6
	Quattro Pro	51,00	6
	SAS	10,00	32

Źródło: Opracowanie własne na podstawie [12]
Objaśnienia: SPR - Software Productivity Research; LOC - Line of Code; FP - Function Point

Model COCOMO

Model COCOMO (ang. *COnstructive COst MOdel*) Barrego Boehma [5] dzieli systemy informatyczne na trzy grupy:

- ***organiczne***, tj. małe projekty informatyczne realizowane przez niewielkie zespoły w dość swobodnie kształtowanych warunkach realizacji; wielkość systemu nie przekracza 50 KDSI (ang. *KDSI - KDelivered Source Instructions*);
- ***półsamodzielne*** (cechy pośrednie między organicznymi a zamkniętymi, wielkość do 300 KDSI);
- ***zamknięte*** (systemy o ostrych ograniczeniach i uwarunkowaniach sprzętowych, programowych, sterowania, itp.).

W zależności od typu systemu informatycznego (w aspekcie powyższych grup) zmieniają się współczynniki modelu [5]. Analogicznie do innych modeli ekstrapolacyjnych ostateczna pracochłonność projektu jest wyznaczana z uwzględnieniem 25 różnych warunków realizacji projektu [5].

Pracochłonność a kalendarzowy czas realizacji projektu

Wszystkie przedstawione dotychczas modele pozwalały wyznaczyć pracochłonność projektu. Przełożenie pracochłonności projektu na terminy jego realizacji wymaga prostego zdawałoby się przejścia od roboczogodzin, osobodni lub osobomiesięcy do konkretnej liczby dni kalendarzowych z uwzględnieniem liczebności zespołu. Jednak przejście to nie jest łatwe. Jak pokazuje praktyka, nie można przyjąć prostej formuły polegającej na podzieleniu pracochłonności przez liczbę pracowników.

Z jednej strony bowiem dość trudno jest oszacować czas pracownika, który może przez niego być poświęcony na realizację projektu. Czas ten bowiem zależy nie tylko od liczby dni roboczych w miesiącu. W oszacowaniach należy uwzględnić straty czasu roboczego na czynności związane lub nie związane z projektem. Typowe straty czasu związane z realizacją projektu, to: podróże, spotkania, komunikacja w zespole, problemy z dostępem do wspólnych zasobów, powtórne wykonywanie prac, bezwładność pracowników (krzywa uczenia się) i brak motywacji pracowników. Znacznie więcej czasu pracownicy tracą na: spotkania nie związane z projektem, opóźnianie rozpoczęcia dnia roboczego i jego skracanie, choroby, telefony, sprawy osobiste, nałogi (palenie), udział w innych projektach, dodatkowe prace na swoim stanowisku pracy (np. sprzątanie, porządkowanie dokumentów, itp.), a także dokształcanie.

Jeśli uwzględnimy wszystkie powyższe czynniki, to się okaże, że musimy liczyć się z nawet 50% utratą czasu roboczego.

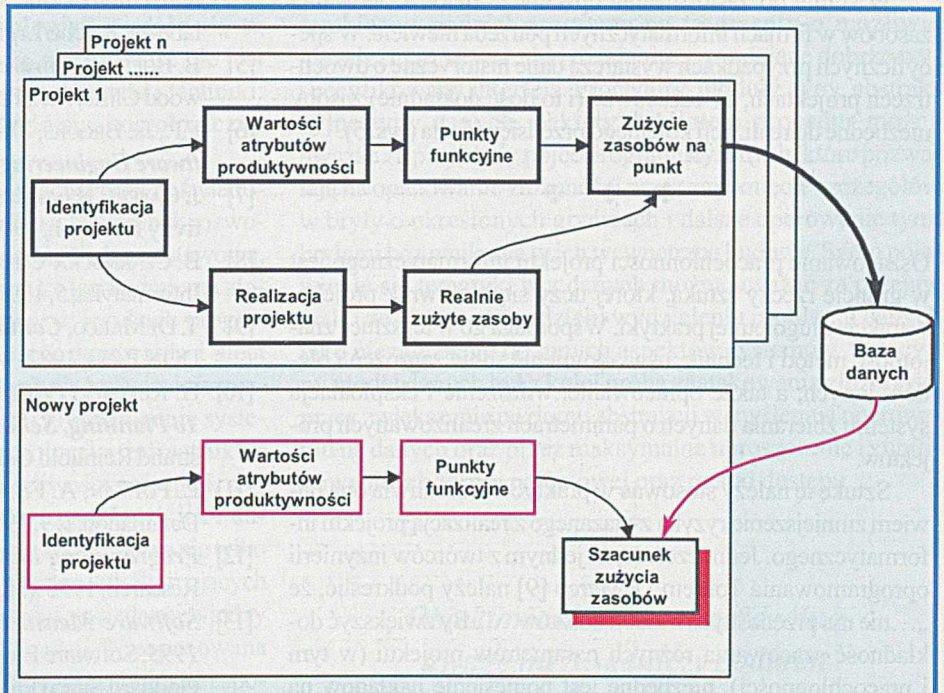
W złożonych projektach informatycznych, realizowanych przez duże zespoły ludzkie, największym problemem, prowadzących do zmniejszenia wydajności pracy, jest komunikacja w zespole. Bardzo obrazowo przedstawia to prawo Brooksa [5]: ***„dodanie nowych pracowników do opóźnionego projektu zwiększa opóźnienie"***. Przyczyn należy szukać w zwiększeniu strat czasu na komunikację pomiędzy członkami zespołu z jednej strony i niepodzielnością pewnych prac- z drugiej.

Istnieje oczywiście pewna „optymalna" wielkość zespołu dla danego projektu. Według metody firmy IBM liczebność zespołu zmienia się pierwiastkowo w zależności od pracochłonności projektu.

W praktyce do określenia potrzeb w liczebności zespołu ludzkiego wykorzystuje się metody dekompozycyjne wspomagane odpowiednimi narzędziami komputerowymi (techniki: CPM, PERT, Gantta, itp.).

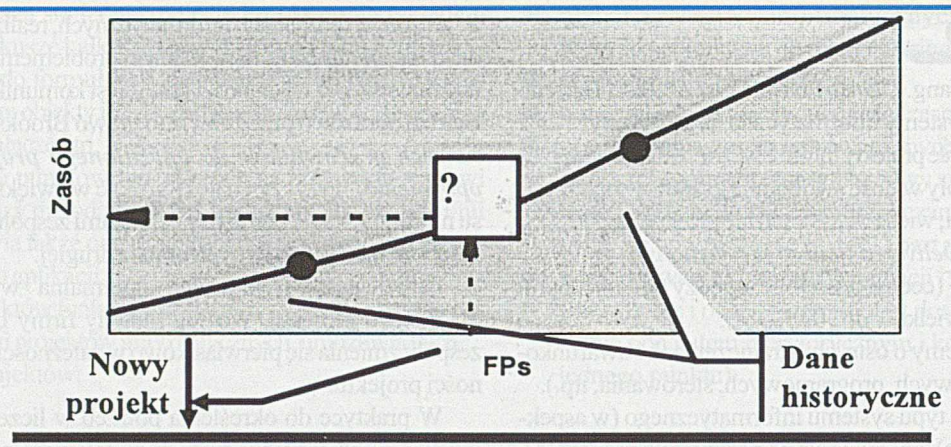
System wspomagający szacowanie pracochłonności projektów

Żaden z przedstawionych modeli oceny pracochłonności i innych parametrów projektów informatycznych nie może być w prosty sposób skopiowany. Zespoły bowiem działają w różnych warunkach, mają różne doświadczenie, wsparcie technologiczne, dostęp do zasobów, itd. Należy przy tym pamiętać, iż technologia informatyczna zmienia się tak szybko, że żadne modele nie są w stanie nadążyć za tymi zmianami. W związku z powyższym, jedynym środkiem podwyższającym jakość szacowania parametrów systemów informatycznych jest wprowadzenie obowiązku wyznaczania (i archiwizowania) wartości atrybutów produktywności na wszystkich etapach normalnej pracy, kolekcjonowania danych i wykorzystania nowych pro-



Źródło: Opracowanie własne

Rys. 4. Schemat systemu wspomagającego szacowanie zasobów na wczesnych etapach realizacji projektu informatycznego



Źródło: Opracowanie własne

Rys. 5. Szacowanie zasobów nowego projektu, niezbyt odbiegającego od już zrealizowanych

jektów do poprawy jakości estymacji. Schemat działania takiego systemu przedstawia rys. 4.

Niewątpliwie warto jest w przedsiębiorstwie zorganizować i ustanowić pewien system wyznaczania pracochłonności zrealizowanych systemów. Odpowiednio opracowana i utrzymywana baza danych o projektach i ich realizacji będzie stanowić doskonałą kopalnię wiedzy dla modeli szacujących pracochłonność budowy całkiem nowych projektów systemów.

W sumie do zastosowania opisanych metod szacowania zasobów w firmach informatycznych potrzeba niewiele. W specyficznych przypadkach wystarczą dane historyczne o dwóch-trzech projektach, by oszacować (i to dość dokładnie) zasoby niezbędne do realizacji kolejnego przedsięwzięcia (rys. 5).

❖ ❖ ❖

Oszacowanie pracochłonności projektu informatycznego jest w gruncie rzeczy sztuką, której uczy się kierownik projektu w trakcie długoletniej praktyki. Wspomaga go w tej sztuce znajomość metod i technik, uświadomienie sobie zagrożeń i błędów innych, a także opracowanie, wdrożenie i eksploatacja systemu zbierania danych o parametrach zrealizowanych projektów.

Sztukę tę należy stosować w praktyce. Umożliwia ona bowiem zmniejszenie ryzyka związanego z realizacją projektu informatycznego. Jednocześnie, za jednym z twórców inżynierii oprogramowania Tomem DeMarco [9] należy podkreślić, że „...nie ma przenośnych modeli kosztów...". By zwiększyć dokładność szacowania różnych parametrów projektu (w tym i pracochłonności), niezbędne jest poniesienie nakładów na stworzenie systemu rejestrowania danych o zrealizowanych projektach i wykorzystywania tej informacji w praktyce.

Literatura:

[1] A. Abran, P.N.Robillard, ***Function Points Analysis: An Empirical Sudy of its Measurement Processes***, IEEE Transactions on Software Engineering, vol. 22, no. 12, 895-909, 1996.

[2] A.J. Albrecht, ***Measuring Application Development Productivity***, Proceedings of the Joint SHARE/GUIDE Application Development Symposium, Chicago, GUIDE International Corp., 1979.

[3] J.W. Bailey, V.R. Basili, ***A Meta-Model for Software Development and Resource Expenditures***, Proceedings of the 5th International Conference on Software Engineering, NY, Institute of Electrical and Electronics Engineers, 1983.

[4] A. Baran, A., P.N. Robillard, ***Function Point Analysis: An Empirical Study of its Measurement Processes***, Montreal, Univ. of Quebec, 1996 (http://saturne.info.uqam.ca/labo-recherche/Lrgl/publi/rjournal).

[5] B. Boehm, ***Software Engineering Economics***, NY, Englewood Cliffs, N.J., Prentice-Hall, 1981.

[6] F.P., Jr. Brooks, ***The mythical man-month: Essays of Software Engineering***; NY, Addison-Wesley Pub. Co., 1995.

[7] J. Capers, ***What Are Function Points?***, Software Productivity Research, 1995 (http://www.spr.com).

[8] B. Czarnocka-Chrobot, ***Metoda FUNCTION POINTS***, Informatyka, 5, 1994, 19-24.

[9] T. DeMarco, ***Controlling Software Projects***, NY, YOURDON Press, 1982.

[10] H. Kerzner, ***Project Management: A System Approach To Planning, Scheduling and Controlling***; NY, Van Nostrand Reinhold Company, 1984.

[11] L. Putman, A. Fitzsimmons, ***Estimation Software Costs***, Datamation, v. 9, 1979, 89-98.

[12] ***Programming Languages Table***, Software Productivity Research, 1996 .(http://www.spr.com/library/langtbl.html).

[13] ***Software Measurement Guidebook. Revision 1, June 1995***; Software Engineering Laboratory Series, Greenbelt, Goddard Space Flight Center, NASA, 1995.

[14] C.E. Walston, C.P Felix, ***A Method of Programming Measurement and Estimation***, IBM Systems Journal., v. 16, 1977, 54-73.

[15] E. Yourdon, ***Współczesna analiza strukturalna***, Warszawa, WNT, 1996.

Marek Miłosz jest pracownikiem Katedry Informatyki Politechniki Lubelskiej.
e-mail: marekm@pluton.pol.lublin.pl

Obiektowość w bazach danych
– koncepcje, nadzieje i fakty

Część I

Kazimierz Subieta

Obiektowość jest nową informatyczną ideologią, która zdobywa coraz liczniejsze grono zwolenników oraz coraz częściej jest obecna w produktach przemysłu informatycznego, w literaturze informatycznej, programach nauczania i działalności naukowej. Przeładowanie terminu „obiektowy" powoduje częste wątpliwości, czym naprawdę jest obiektowość i jakie to ma konsekwencje dla obecnych i przyszłych technik, języków i systemów. Szczególnie gorąca debata dotyczy baz danych, ze zrozumiałego powodu: w dziedzinie tej zarabia się ogromne pieniądze, w skali globalnej miliardy dolarów rocznie. Emocje rozgrzewa także fakt, że obiektowość spycha do cienia poprzednią wielką bazodanową ideologię, zwaną modelem relacyjnym.

W istocie, obiektowość jest manifestacją hipotezy dotyczącej pożądanego kierunku rozwoju oprogramowania i technik jego tworzenia. Termin „ideologia" wydaje się tu właściwy również z tego powodu, że obiektowość nie jest wyłącznie zjawiskiem odnoszącym się do nauki i techniki: ma ona również wymiar psychologiczny, socjologiczny i kulturotwórczy.

Jak wszystkie ideologie, obiektowość wynika ze świadomości wad istniejącej rzeczywistości i ustala kierunek rozwoju, którego celem jest usunięcie tych wad. Wadą podstawową, brzemieniem obecnego czasu informatyki, jest nadmierna złożoność. Złożoność jest wszechobecna: w językach programowania, w konstrukcji sprzętu, oprogramowania i sieci informatycznych, w strukturach danych, w świecie opisywanym przez dane, w metodach analizy i projektowania systemów, itd. Złożoność obecnych ontologii i infrastruktur informatycznych często przerasta możliwości percepcji, wyobraźni i organizacji zarówno pojedynczych ludzi, jak i dużych zespołów ludzkich. Konsekwencją złożoności są nie tylko frustracje większości informatyków i użytkowników, mających stałe uczucie kruchości i tymczasowości posiadanej wiedzy oraz niepewności podejmowanych decyzji. Nieopanowana złożoność ma konsekwencje techniczne, ekonomiczne i społeczne w postaci czasu i kosztów tworzenia produktów informatyki, ich jakości, szkolenia, praktycznej użyteczności systemów, ich niezawodności, modyfikowalności. Nadmierne koszty, bezużyteczne systemy, błędy w oprogramowaniu krytycznych urządzeń technicznych, ograniczenia funkcjonalności są często konsekwencją tego, że twórcy systemów informatycznych nie potrafili opanować nakładającej się, spiętrzonej złożoności różnych ich aspektów.

Obiektowość jest nową nadzieją na zredukowanie złożoności. Nie jest to pierwsza ideologia o tak sformułowanej misji. Na przełomie lat 60. i 70. taką ideologią było programowanie strukturalne. Jego podstawowym założeniem był hierarchiczny podział programu na fragmenty o możliwie niezależnym znaczeniu i działaniu, posiadające dobrze wyspecyfikowany interfejs. Procedury, moduły, typy, abstrakcyjne typy danych, pakiety, biblioteki procedur można uważać za przykłady pojęć programistycznych, które pozwalają na opanowanie złożoności przez zamknięcie szczegółów w bryły o określonych granicach i dalsze operowanie tymi bryłami bez wnikania w ich wewnętrzną budowę. Samo pojawienie się tematyki baz danych można uważać za przejaw walki ze złożonością, dzięki wydzieleniu problemu danych jako niezależnego od innych aspektów systemów. Relacyjny model danych był z kolei próbą zaatakowania złożoności przez zwiększenie poziomu abstrakcji w myśleniu i operowaniu na danych oraz przez maksymalne uproszczenie i zunifikowanie ich formy pojęciowej oraz metod dostępu.

Obiektowość zwraca szczególną uwagę na zasadniczy podmiot w produkcji oprogramowania, jakim jest czynnik ludzki.

Wcześniejsze ideologie komputerowe były w dużym stopniu obciążone kryterium, w którym dobrem podstawowym była wydajność komputera rozumiana jako szybkość działania systemu i jego zapotrzebowanie na różne rodzaje pamięci. Tych czynników nie można lekceważyć; niemniej wśród obecnych problemów czynnik ludzki staje się dominujący. Jest on

zasadniczy dla szybkości tworzenia produktów informatycznych oraz dla ich cech użytkowych, takich jak: jakość, uniwersalność, elastyczność, niezawodność, skuteczność, estetyka, które na równi z wydajnością stają się kluczowe dla produktów współczesnej informatyki.

Cel nadrzędny obiektowej ideologii polega na lepszym dopasowaniu modeli pojęciowych i modeli realizacyjnych systemów do wrodzonych instynktów, własności psychologicznych i mentalnych mechanizmów percepcji i rozumienia świata. J. Martin pisze: „Mój kot jest zorientowany obiektowo", co ma uzmysłowić fakt, że miliony lat ewolucji wykształciło mechanizmy percepcji, których podstawą jest wyróżnianie w otaczającym nas świecie obiektów o określonych granicach, które mają odpowiedniki językowe w postaci rzeczowników lub fraz rzeczownikowych i podlegają określonemu zachowaniu lub manipulacjom. Zwrot ku obiektowości w informatyce jest zjawiskiem pozytywnym głównie ze względu na to, że centralnym punktem zainteresowań staje się nie narzędzie w postaci komputera lub systemu komputerowego, lecz człowiek (projektant, programista, użytkownik) z jego wrodzoną percepcją otaczającego go świata i odpowiadającą mu wizją wnętrza systemu informatycznego. Nawet w przypadku, kiedy to wnętrze nie ma bezpośrednich skojarzeń ze światem zewnętrznym, projektant i programista buduje w swoim umyśle obrazy tworzonych i przetwarzanych danych, które mają podobne cechy jak obrazy obiektów materialnych. Obiektowość - w założeniu - bazuje na własnościach i mechanizmach ludzkiego umysłu na podobnej zasadzie jak koncepcja klawiatury komputera bazuje na anatomicznym fakcie posiadania przez ludzi rąk i palców.

Dla realizacji obiektowości zaproponowano ogromną kolekcję pojęć, metod, narzędzi, języków i systemów. Obiektowość jako nazwana ideologia liczy już kilkanaście lat: za początek jej intensywnego rozwoju obiektowości można uznać rok 1980, kiedy pojawił się język Smalltalk. Od tego czasu obiektowość przeniknęła ogromne obszary informatyki, włączając takie dyscypliny, jak: języki programowania, bazy danych, sieci komputerowe oraz metodyki analizy i projektowania systemów. Spontaniczny rozwój obiektowości zaowocował oceanem wiedzy, w którym nawet ambitni specjaliści poruszają się z wielkim trudem. Informatyka, jak chyba żadna inna dziedzina, napędza się przede wszystkim coraz to nowszymi wynalazkami, zaś w tej powodzi wynalazków dość często nie wystarcza czasu i zapału do budowania syntetycznych zasad, teorii, klasyfikacji, spójnych systemów pojęć. Z tego powodu wiedza o obiektowości jest w dużym stopniu pochodną ideologicznych założeń, teorii o podłożu intelektualno-estetycznym, myślenia życzeniowego i (przede wszystkim) faktów: metod, języków, bibliotek, systemów, standardów, metodyk. W serii trzech artykułów postaramy się zrekapitulować tematy, które uważamy za istotne dla tematyki obiektowości w bazach danych, zwracając szczególną uwagę na istotne fakty dotyczące rozwoju obiektowości w strefie przemysłowo-komercyjnej. Zainteresowanych odsyłamy do bardziej wyczerpujących pozycji, np. [1], [2], [3], [4], [5]. Wyjaśnienia wielu terminów dotyczących obiektowości można znaleźć w [6]; słownik ten jest również dostępny poprzez WWW.

Oczekiwania dotyczące obiektowości

Obiektowość wyrasta na gruncie poczucia niedoskonałości obecnego stanu rzeczy w zakresie oprogramowania, które określa się mianem „kryzys oprogramowania". Dość często podkreślane są następujące znamiona tego kryzysu:

- Sprzeczność pomiędzy ogromną odpowiedzialnością, jaka spoczywa na współczesnych systemach informatycznych, a ich zawodnością wynikającą ze złożoności i niedojrzałych metod tworzenia i weryfikacji oprogramowania. Pamiętny apel Davida Parnasa przypomina o dramatycznych konsekwencjach (do wybuchu wojny atomowej włącznie), jakie ludzka cywilizacja może ponieść wskutek powierzenia jej losów zawodnym systemom komputerowym;
- Ogromne koszty wytwarzania i utrzymania systemów informatycznych. Dla przykładu, utrzymanie 10 miliardów linii kodu kosztuje szacunkowo w USA 70 miliardów dolarów rocznie (*IEEE Software Development, Aug 94, p.65*);
- Długi cykl tworzenia oprogramowania, niska wydajność twórców oprogramowania; eklektyczny, nieregularny, niesystematyczny i zbyt niski poziom zestawu narzędzi i języków programowania;
- Wysokie prawdopodobieństwo niepowodzenia projektu programistycznego;
- Frustracje projektantów oprogramowania i programistów wynikające ze zbyt szybkiego postępu w zakresie języków, narzędzi i metod oraz zbytniego odhumanizowania procesów produkcji oprogramowania; zbyt wczesne starzenie się twórców oprogramowania;
- Ogromne problemy, jakie niesie za sobą bezwładność wdrożonych systemów informatycznych, zbytnie uzależnienie organizacji od systemów komputerowych, problemy współdziałania niezależnie zbudowanych systemów (szczególnie istotne przy obecnych tendencjach integracji różnych systemów), problemy systemów „spadkowych" (*legacy*), czyli przystosowanie starych funkcjonujących systemów do nowych wymagań, tendencji i platform sprzętowo-programowych.

Obiektowość stwarza nadzieję na opanowanie kryzysu oprogramowania przez zwiększenie wydajności twórców oprogramowania, zwiększenie poziomu abstrakcji i dzięki krótszemu cyklowi tworzenia oprogramowania i łatwiejszej jego modyfikacji.

Uważa się, że efekty te są konsekwencją następujących generalnych własności, takich jak:

- **Mechanizmy abstrakcji** pozwalające budować abstrakcyjne struktury i operować tymi strukturami bez wnikania w ich wewnętrzną budowę. Oprócz klasycznych

procedur, modułów i typów takie mechanizmy zapewniają klasy, dziedziczenie, metody, hermetyzacja, późne wiązanie i polimorfizm;

- **Mechanizmy kompozycji i dekompozycji** pozwalające grupować struktury w coraz większe bryły o dobrze zdefiniowanych charakterystykach; z drugiej strony, pozwalające dekomponować złożone struktury na mniejsze dobrze zdefiniowane fragmenty i rozpatrywać te fragmenty niezależnie od całości i od pozostałych fragmentów;
- **Ponowne użycie** (*reuse*) wcześniej wytworzonych komponentów projektów i oprogramowania. Może ono dotyczyć wszystkich elementów projektu, ale szczególne znaczenie ma definiowanie klas ponownego użycia oraz mechanizmy hermetyzacji i generalizacji/specjalizacji (z przesłanianiem, późnym wiązaniem i polimorfizmem). Obiektowość zaowocowała wieloma pojęciami dotyczącymi ponownego użycia, takimi jak np. wzorce projektowe, szablony (*templates*), metaklasy i klasy parametryczne.

Doskonalenie mechanizmów abstrakcji, kompozycji/dekompozycji i środków ponownego użycia stwarza potencjał zwiększający jakość, niezawodność, testowalność, rozszerzalność, współdziałanie, łatwą pielęgnację oprogramowania, szybkie prototypowanie oraz dobrą dokumentację. Ten wniosek potwierdzają liczne obserwacje dotyczące rzeczywistych projektów.

Nadzieje, które niesie obiektowość, są związane z wyeliminowaniem niektórych wad tradycyjnych metod analizy i projektowania. Jedną z nich było słabe uwzględnienie w analizie i projekcie cech proceduralnych. Klasyczne metodyki analizy i projektowania systemów były osnute wokół dwóch trzonów: diagramów przepływu danych oraz diagramów modelujących statyczny stan obiektów w obserwowanej rzeczywistości i powiązania pomiędzy tymi obiektami (np. diagramy encja-związek). Obydwa paradygmaty nie uwzględniały operacji przypisanych do obiektów, co powodowało, że były one elementem mniej systematycznym, umykającym z pola uwagi analityków i projektantów. W istocie te elementy pojawiały się w późniejszych fazach realizacyjnych, na etapie implementacji, tworzenia modułów systemowych i bibliotek procedur. W tym zakresie obiektowość wnosi rewolucyjną zmianę: elementy te zostały przesunięte zarówno do wcześniejszej fazy projektu, jak i do innego repozytorium. Operacje (zwane metodami) są uwzględniane łącznie z klasami obiektów już na najbardziej wczesnym etapie analizy pojęciowej, co umożliwia znacznie bardziej systematyczne ich potraktowanie, bardziej precyzyjną specyfikację poszczególnych etapów projektu oraz wyeliminowanie rożnorodnych ich anomalii, np. redundancji, zbytniej uniwersalności metod lub zbytniej ich specjalizacji. Z drugiej strony, operacje są przesunięte ze słabo uporządkowanych bibliotek i modułów aplikacyjnych do klas, co zwiększa zarówno stopień abstrakcji, jak i możliwości kompozycji/dekompozycji; jest to również istotny potencjał dla ponownego użycia.

Podstawowe założenia obiektowości

Jest dość trudno wydzielić powszechnie uznane założenia obiektowości; jest ona bowiem doktryną ideologiczną, czyli przedmiotem wiary, niewiary i zróżnicowanych poglądów co do wagi i definicji poszczególnych jej cech. Niżej podajemy pewien typowy ich zestaw.

- **Złożone obiekty, tożsamość**. Oprogramowanie i przechowywane dane powinny składać się ze zrozumiałych modułów zawierających struktury danych z przypisanymi do nich operacjami, czyli *obiektów*. Obiekty mogą być dowolnie duże i dowolnie złożone, ale reguły operowania obiektami nie powinny od tego zależeć. Obiekty mają przypisaną tożsamość, co oznacza, że istnieją i są identyfikowalne niezależnie od ich aktualnego stanu lub miejsca przechowywania.

- **Powiązania** (*relationships*). Obiekty mogą być powiązane związkami asocjacyjnymi, np. związek *JestZatrudnionyW* łączy obiekty *Pracownik* z obiektami *Firma*. Powiązania są w sposób naturalny odwzorowane w strukturze danych, np. w postaci wskaźników prowadzących od obiektu do obiektu.

- **Hermetyzacja** (*encapsulation*). Oznacza rozróżnienie pomiędzy interfejsem do obiektu opisującym *co* obiekt zawiera i *co* robi, a implementacją definiującą *jak* on jest zbudowany i *jak* on to robi. Hermetyzacja oznacza także ukrywanie informacji niepotrzebnej na danym etapie projektowania lub programowania.

- **Klasy, typy, interfejsy**. Obiekty są klasyfikowane zgodnie z podobieństwem ich charakterystyk i użycia. Cechy niezmienne dla tych obiektów, w szczególności zestaw ich atrybutów (wraz z ich typami) oraz operacje, które można na nich wykonać (metody), są zbierane wewnątrz specjalnego bytu programistycznego, który nazywa się klasą. Typy umożliwiają statyczną i dynamiczną kontrolę poprawności budowy obiektów, poprawności ich użycia w programach oraz poprawności użycia operacji przypisanych do obiektów. Obiektowość zakłada oddzielenie zewnętrznej specyfikacji klasy, zwanej interfejsem, od jej implementacji.

- **Metody i komunikaty**. Wszelkie operacje na obiektach wykonuje się przy pomocy metod, czyli procedur wykonywanych w środowisku wnętrza obiektu. Obiekt wykonuje jedną z przypisanych do niego operacji po wysłaniu komunikatu zawierającego jej nazwę (oraz parametry). Komunikat taki nie zależy od szczegółów implementacji lub reprezentacji obiektu.

- **Hierarchia klas i dziedziczenie**. Klasy są organizowane w hierarchię (lub inną strukturę) zakresów znaczeniowych; klasy bardziej szczegółowe dziedziczą niezmienniki klas bardziej ogólnych (definicje atrybutów, metody, definicje zdarzeń lub wyjątków i inne). Konsekwencją tego założenia jest możliwość skorzystania z klas bardziej ogólnych przy definiowaniu klasy będących ich specjalizacją: nowa klasa zawiera wszystkie wcześniej zdefiniowane cechy plus niektóre nowe cechy. Z drugiej strony, daje to możliwość utworzenia klasy abstrakcyj-

nej, tj. „wyciągnięcia przed nawias" (*factoring out*) wspólnych cech pewnej grupy klas i utworzenia z nich nowej klasy.

- **Przesłanianie, późne wiązanie, polimorfizm**. Wybór nazwy dla operacji jest określony wyłącznie jej zewnętrznym, pojęciowym znaczeniem w ramach danej klasy obiektów; wybór ten nie jest uwarunkowany jakimikolwiek własnościami lub istnieniem innych klas. Ten sam komunikat wysłany do różnych obiektów może wywoływać różne operacje; tę własność określa się jako polimorfizm. Własność ta wymaga wiązania nazw na etapie wykonania (późnego wiązania). Przesłanianie (*overriding*) oznacza, że metoda odziedziczona przez obiekt z klasy bardziej wyspecjalizowanej przesłania metodę o tej samej nazwie odziedziczoną z klasy bardziej ogólnej.

- **Trwałość** (*persistence*). Niektóre obiekty zachowują swój stan dłużej niż czas pojedynczego uruchomienia programu (zwykle, dowolnie długo). Trwałość jest podstawową własnością obiektów przechowywanych w bazie danych.

Obiektowy model danych

Często stwierdza się, że model relacyjny był odpowiedni dla zastosowań tradycyjnych takich jak zastosowania bankowe, ale jest nieadekwatny dla nowoczesnych zastosowań w takich dziedzinach, jak: CAD (*Computer Aided Design*), CAM (*Computer Aided Manufacturing*), CASE (*Computer Aided Software Engineering*), multimedia, OLAP (*On Line Analytical Processing*), itd. Zważywszy na wyposażenie współczesnych systemów relacyjnych i postrelacyjnych w takie cechy, jak: możliwość pracy z multimediami (BLOBy), możliwość umieszczania dowolnie długich ciągów znaków zmiennej długości jako wartości atrybutów relacji, itp., teoretycznie nie ma przeszkód w zaimplementowaniu dowolnego zastosowania z wyżej wymienionych dziedzin. Istotną zaletą modelu obiektowego jest jednak wyższy poziom abstrakcji, który umożliwia zaprojektowanie i zaprogramowanie tej samej aplikacji w sposób bardziej skuteczny, konsekwentny i jednorodny. Argumentacja na rzecz modelu obiektowego stawia więc na uproszczenie i usystematyzowanie procesu projektowania i programowania, minimalizację liczby pojęć, zwiększenie poziomu abstrakcji, zmniejszenie dystansu pomiędzy fazami analizy, projektowania i programowania oraz zwiększeniu nacisku na rolę czynnika ludzkiego.

Model obiektowy dotyczy głównie struktur danych przechowywanych w obiektowej bazie danych. Wyznacza on bazę intelektualną i pojęciową określającą budowę struktur danych oraz komunikację pomiędzy ludźmi. W stosunku do modelu relacyjnego obiektowość wprowadza znacznie więcej pojęć, których znaczenie nie zawsze jest jednakowo rozumiane. Z tego powodu stworzenie powszechnie akceptowanego modelu obiektowego jest sprawą trudną; temat ten jest przedmiotem prac kilku ciał standaryzacyjnych, w szczególności OMG i ODMG.

Filarami, na których opiera się każdy model obiektowy, są omówione wcześniej pojęcia: złożone obiekty, tożsamość, powiązania, klasy i typy, hierarchia (lub inna struktura) dziedziczenia, metody, komunikaty, hermetyzacja, przesłanianie, późne wiązanie, polimorfizm i trwałość.

Model obiektowy precyzuje sposób rozumienia wymienionych pojęć oraz ustawia je w pewien system, który (w założeniu) ma być wewnętrznie spójny. Taki model jest podstawą ideologiczną architektury systemów, języków, z którymi mają do czynienia różne klasy ich użytkowników oraz interfejsów programistycznych. Bazy danych mają do czynienia z następującymi elementami zależnymi od modelu:

- **Architekturą systemu zarządzania bazą danych**. Wyznacza ona najbardziej ogólne założenia dotyczące składowych systemu, poziomów abstrakcji w organizacji systemu, funkcjonalności systemu oraz różnorodnych powiązań pomiędzy składowymi i/lub funkcjami systemu.

- **Strukturami danych podtrzymywanymi przez system**. W przypadku systemów obiektowych takimi strukturami są obiekty, atrybuty i związki asocjacyjne podlegające grupowaniu w klasy, dziedziczeniu inwariantów i hermetyzacji. To oczywiście nie wyczerpuje tematu ze względu na dalsze założenia co do stopnia uniwersalności wymienionych cech oraz (np. możliwość tworzenia obiektów o strukturze hierarchicznej z polami wielowartościowymi) ze względu na kombinację z innymi pojęciami i założenia co do dynamicznych własności tych struktur (np. przenoszenia obiektów do innych klas, dostawianie i usuwanie atrybutów, itd.)

- **Językiem opisu danych** (*Data Description Language, DDL*). W języku tym projektant bazy danych zapisuje jej strukturę, która niezbędna jest do rozumienia semantyki danych, a także sposobów, w jaki ta semantyka została odwzorowana w postaci zapamiętanych struktur danych. Wyrażenie języka opisu danych nosi nazwę schematu. Schemat specyfikuje przechowywane dane na pewnym poziomie abstrakcji (poziom logiczny) poprzez podanie ich typów, klas, powiązań klas w hierarchie dziedziczenia, powiązań asocjacyjnych i innych elementów.

- **Językiem manipulacji danymi** (*Data Manipulation Language, DML*), za pomocą którego programista uzyskuje możliwość wyszukiwania danych w bazie danych oraz ich przetwarzania i aktualizacji. DML uważa się za pewien ograniczony język programowania aplikacji z baz danych, który w zasadzie wymaga zanurzenia go w pewien uniwersalny język programowania taki jak C++. Istnieje tendencja poszerzania funkcji DML, co często prowadzi do wyodrębnienia go jako samodzielnego języka programowania.

- **Językiem zapytań** (*Query Language*), za pomocą którego programista wyszukuje dane oraz dokonuje pewnych (zwykle prostych) operacji na danych. Język zapytań może być rozpatrywany jako język interakcyjnej pracy użytkownika z systemem. Częściej jednak język zapytań jest uważany za składową języka DML lub wręcz za synonim DML. Przy takim zastosowaniu język zapytań jest zestawem konstrukcji wysokiego poziomu, który może być zintegrowany z uniwersalnym językiem programowania. Ponieważ nie istnieje granica, gdzie „kończy się" język zapytań, a „zaczyna się" język programowania, istnieje tendencja poszerzania możliwości języka zapytań w kierunku języka programowania; np. niektóre języki czwartej generacji (*4GL*), PL/SQL systemu Oracle lub SQL3, będące w założeniu pełnymi językami programowania.

Istnieje wiele innych pojęć, które mogą być składnikiem lub konsekwencją obiektowego modelu danych, np. obiektowe perspektywy (*object views*), wartości zerowe (*null values*), unie (*unions*), wersje, katalogi danych, aktywne reguły (*active rules*) i inne.

W kolejnych częściach, które ukażą się w następnych numerach, przedstawimy przegląd ważniejszych kierunków badawczo-rozwojowych w zakresie obiektowych baz danych oraz dokonamy porównania obiektowości z modelem relacyjnym.

Literatura

Większość materiału pochodzi z przeogromnej bazy danych zwanej WWW. Nieocenione usługi oddały autorowi serwisy AltaVista (www.altavista.com), HotBot (www.hotbot.com) i Yahoo (www.yahoo.com). Wyliczenie wszystkich źródeł WWW, z których autor korzystał przy pisaniu tego artykułu (i innych) jest przedsięwzięciem beznadziejnym: były ich setki, o ile nie tysiące. Podobnie jest z materiałami drukowanymi – niżej zamieszczony jest mały wyciąg z setek publikacji.

[1] D. Figura. ***Obiektowe bazy danych***. Akademicka Oficyna Wydawnicza, Warszawa ,1996;

[2] W. Kim. ***Wprowadzenie do obiektowych baz danych***. Wydawnictwa Naukowo-Techniczne, Warszawa, 1996;

[3] M.E.S. Loomis. ***Object Databases: The Essentials***. Addison Wesley, 1995;

[4] A.R. Simon. ***Strategic Database Technology: Management for the Year 2000***. Morgan Kaufmann, 1995;

[5] S.B. Zdonik, D. Maier. ***Fundamentals of Object-Oriented Databases***. In S.B. Zdonik and D. Maier (Eds.), Readings in Object-Oriented Database Systems, Morgan Kaufman, 1990, pp.1-32;

[6] K. Subieta. ***Słownik często spotykanych terminów dotyczących obiektowości***. Prace IPI PAN Nr 839, wrzesień 1997. Postscript dostępny poprzez: http://www.ipipan.waw.pl/~subieta

Kazimierz Subieta jest kierownikiem zespołu inżynierii oprogramowania w Instytucie Podstaw Informatyki PAN oraz wykładowcą w Polsko-Japońskiej Wyższej Szkole Technik Komputerowych. e-mail: subieta@ipipan.waw.pl
WWW: http://www.ipipan.waw.pl/~subieta

Komentarz z pozycji obserwatora

Informatyka bankowa w kraju i na świecie

Część II

Zygmunt Ryznar

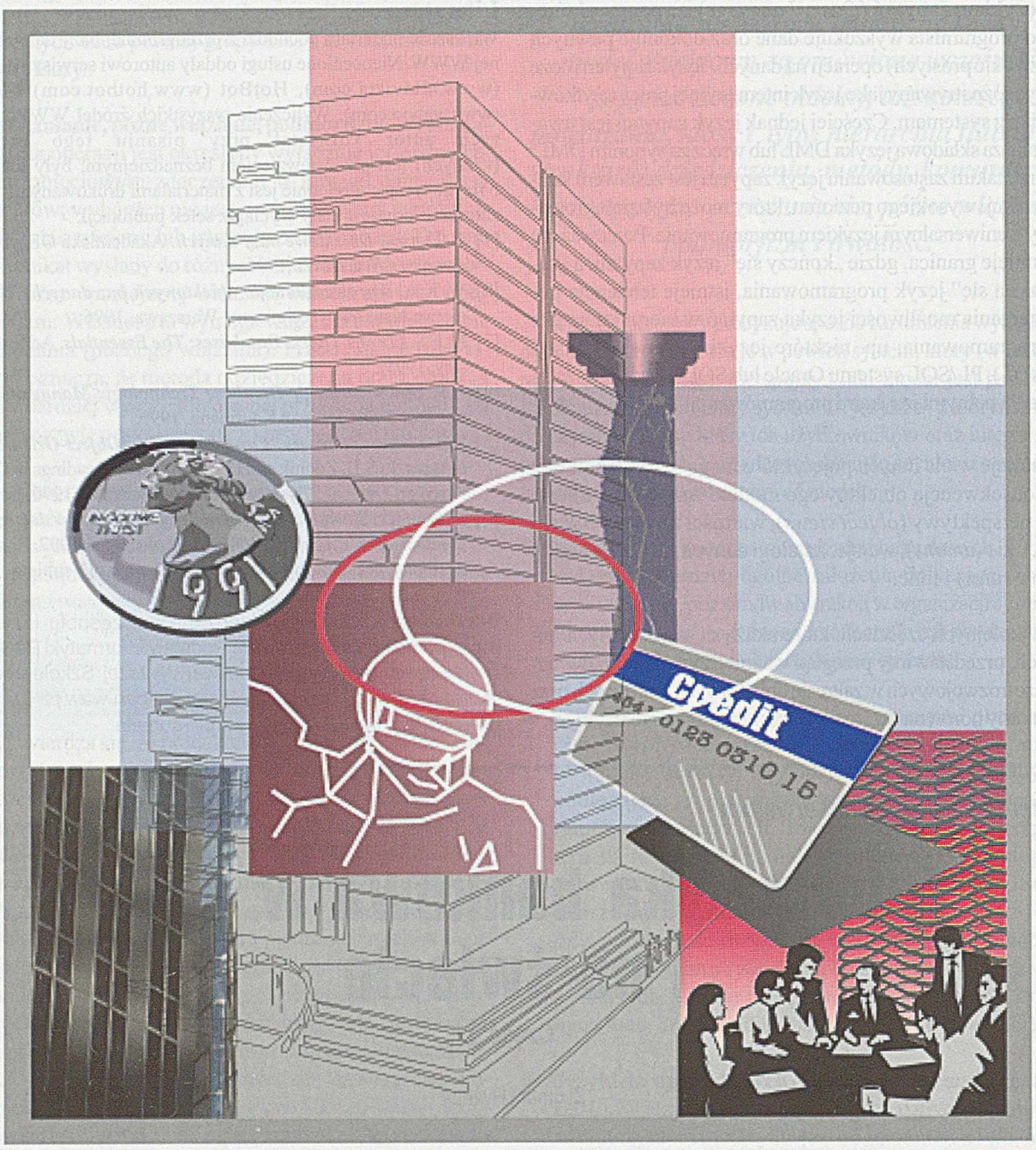
Bitwa o „midway", czyli o duże pieniądze w bankowości

Producenci komputerów do tej pory z reguły nie angażowali się bezpośrednio w tworzenie aplikacji bankowych. Do wyjątków należy firma UNISYS, oferująca od dawna zarówno swój własny sprzęt, jak i swoje własne systemy aplikacyjne.

Banki mają pieniądze i istnieje tam do rozwiązania wiele problemów informatycznych .

Stąd ostatnie lata przyniosły zmianę stanowisk producentów. Firma DIGITAL oferuje system DECbank FBS (Financial Business System) opracowany przez kupioną w 1992 r. od Philipsa szwedzką firmę BCFI AB. System FBS przeznaczony jest do definiowania i obsługi oddziałowych produktów bankowych (branch delivery channels), posiada wbudowany moduł FM (Work Flow Management) i może współpracować z podstawowym (centralnym) systemem bankowym, np. z systemem PROFILE. HP zawarł w 1996 roku kontrakt z Banco Bamerindus (3000 oddziałów) w Brazylii na budowę systemu bankowego w architekturze klient-server (bank ten zrezygnował z systemu opartego na mainframe'ach).

O zmianie strategii IBM świadczy kontrakt (1996 r.) z 15 bankami Ameryki Północnej o utworzeniu firmy, która będzie świadczyć usługi telebankowe dla połowy klientów detalicznych rynku amerykańskiego, opierając się na globalnej sieci IBM. W Europie IBM i grupa finansowych instytucji podjęły prace nad stworzeniem nowoczesnego systemu o nazwie bankowej „Corebank", opartego na IBM-owskich metodach i narzędziach IFW (Information FrameWork) i FSDM (Financial Services Data Model). System przeznaczony jest dla banku detalicznego. .Pracuje na relacyjnej bazie danych DB 2, posiada łącze do Lotus Notes, prowadzi obsługę całodobową w tym transakcji realizowanych przez bankomaty.



Zapowiadany jest jako elastyczny i wielojęzyczny oraz wyposażony w narzędzia użytkownika. Jaki jest naprawdę, pokażą wdrożenia.

Do współpracy z producentami komputerów coraz bardziej angażuje się największa firma software'owa Microsoft, której system operacyjny Windows NT znajduje szerokie zastosowanie w aplikacjach finansowych. Ponadto Microsoft zawarł porozumienie z Hewlett-Packard o wspólnym opracowywaniu systemu bankowego . Prawdopodobnym odbiorcą tego systemu będzie Bangue de France, migrujący do Windows NT na komputerach HP. Również firma Digital zawarła porozumienie (Alliance for Enterprise Computing) z Microsoft, deklarując stosowanie Windows NT na własnych platformach ALPHA, stosowanie wspólnego API (włącznie z DDE). Compaq miał powołać Focused Banking Team do testowania rozwiązań bankowych dla platform opartych na oprogramowaniu Microsoft. O zaangażowaniu firmy Microsoft - w ostatnim okresie - w poszukiwaniu rozwiązań dla bankowości (w sieci internetowej - Merchant System) świadczy powołanie specjalnego pionu pod nazwą Desktop Finance Division. Tak więc Microsoft staje się agresywny nie tylko w dziedzinie narzędziowej, lecz również aplikacyjnej.

Krajowe systemy bankowe [1)]

Duża liczba firm software'owych w Polsce, zajmujących się opracowywaniem systemów bankowych oraz rozrzut tery-

[1)] Uwagi na temat krajowych systemów bankowych oparte są na moich materiałach umieszczonych w opracowaniu: „Krajowe Systemy Bankowe", DiS, Otwock, listopad 1996.

rytorialny tych firm po całym kraju (duże i małe miejscowości), świadczy o inwencji polskich informatyków. W sumie liczba systemów bankowych i autonomicznych modułów jest znaczna (ponad 20) i porównywalna z liczbą w tak rozwiniętym kraju jak Niemcy, co stanowi powód do zastanowienia.

Nie musi bowiem to oznaczać bogactwa rozwiązań i gwarantować dalszego postępu. Przedmiotem oceny bowiem powinna być nie ilość systemów, lecz zakres ich funkcjonalności, jakość rozwiązań (pochodząca w dużej mierze od wiedzy bankowej i używanych narzędzi programistycznych) oraz potencjał firm autorskich.

„Bogactwo" rozwiązań jest poniekąd sztuczne, gdyż oferowane są podobne rozwiązania, co wzmaga wzajemną konkurencyjność, a po drugie, systemy pochodzą od małych firm autorskich, co ogranicza możliwość finansowania rozwoju systemów.

Wśród firm zajmujących się oprogramowaniem jedynie firma PROKOM (4 pozycja w raporcie TOP200 ComputerWorld Polska. „Polski rynek komputerowy i telekomunikacyjny 1996") dysponowała w 1996 r. dużym kapitałem własnym - 86 mln zł (z tym, że oprogramowanie finansowo-bankowe ma niewielki udział w jej obrotach ogólnych). Kolejna(pod względem wielkości obrotów) firma- Softbank posiadała 16 mln zł kapitału własnego, przy przychodach 23 mln zł z oprogramowania finansowo-bankowego. Przychody innych większych firm mieściły się w przedziale 2-8 mln zł.

Utworzona grupa kapitałowa ComputerLandu, który przejął kontrolę nad firmami software'owymi ELBA i CSBI pracującymi dla banków, może oznaczać zmianę tego stanu (oczywiście pod warunkiem, że dojdzie do koordynacji czy też integracji poczynań tych firm, dotychczas konkurujących ze sobą).

Rynek krajowych systemów informatycznych dla bankowości można scharakteryzować w następujący sposób:

a) Duża liczba wdrożeń

Jest ona adekwatna do stanu organizacyjnego polskiej bankowości, w której przeważają małe banki jedno- i kilkuoddziałowe oraz stosuje się rozwiązania zdecentralizowane. Liczby wdrożeń mówią więc głównie o liczbie instalacji oddziałowych, dlatego należy to mieć na uwadze, porównując je ze statystyką wdrożeń umieszczoną w wykazie systemów zagranicznych, gdzie często jedno wdrożenie oznacza system dla całego banku.

Dla małych polskich banków systemy zagraniczne są za drogie i pod względem profilu (obsługa centrali, bankowość komercyjna i operacje zagraniczne) zwykle nie odpowiadają ich potrzebom. Systemy krajowe mają z reguły charakter zdecentralizowany i instalowane są w większości w małych bankach. Z drugiej strony duża liczba wdrożeń potwierdza wiarę twórców w rozwój (w kierunku spełniania aktualnych wymagań odbiorców w celu ich utrzymania jako klientów) oraz wdrożenia i serwisowania tak znacznej ilości zainstalowanych systemów.

b) Stosunkowo zawężony zakres obsługi central banków oraz praktycznie brak tzw. bankowości komercyjnej

Typowy zakres obsługi centrali, proponowany w systemach krajowych, sprowadza się zwykle do agregacji/konsolidacji księgi głównej oraz rozliczeń międzybankowych i międzyoddziałowych. Czasem centralnie prowadzona jest kartoteka klientów oraz rejestr zastrzeżeń (rachunków, czeków, dokumentów, itp.). Należy przewidywać, że zastosowania dla potrzeb centrali będą w przyszłości najbardziej się rozwijać w systemach krajowych. Wynika to ze wzrastającego zapotrzebowania na nowe instrumenty finansowe, wzrostu skali bankowości (konsolidacja banków), potrzeby obsługi zarządzania ryzykiem finansowym i stosunkowo słabego dotychczas zaawansowania prac w tym zakresie. W szczególności upowszechniane muszą być takie dziedziny bankowości komercyjnej, jak: operacje zagraniczne, dokumentowe płatności (w tym akredytywa), operacje na rynku pieniężnym i kapitałowym, itp. Będzie potrzebne nie tylko oprogramowanie obsługi samych transakcji, lecz także do zarządzania portfelami inwestycyjnymi oraz umożliwiające ściślejszą kontrolę przepływu pieniężnego w skali całego banku.

c) Z reguły zaawansowany i różnorodny zakres oddziałowej bankowości detalicznej,

Istniejące rozwiązania bankowości detalicznej są wystarczające do obsługi klientów krajowych w zakresie jednowalutowych, podstawowych produktów (ror, lokaty, proste kredyty) pod względem funkcjonalnym i wymogów czasu rzeczywistego (zwykle obejmuje on salda klienta i generowanie księgowań).

d) Przejmowanie (tworzenie lub implementowanie) pewnych nowości zagranicznych

Mam tutaj na myśli corporate/office banking (usługi bankowe dzięki stacjom roboczym zlokalizowanym w przedsiębiorstwach) zwane u nas w kraju homebankingiem - na Zachodzie pojęcie to oznacza zwykle usługi bankowe inicjowane z komputerów klientów indywidualnych) i pakiety wspomagania informacyjnego kierownictwa oparte na technologii hurtowni danych i OLAP-u.

e) Zaawansowana parametryzacja produktów oddziałowych

Przeważnie nie wymagająca zmian w kodach źródłowych, lecz czasem zbyt skomplikowana dla użytkowników nie będących informatykami.

f) Profesjonalizm programistów

Wyraża się on w tworzeniu narzędzi aplikacyjnych (np. generatorów: raportów, formatek ekranowych, menu), używaniu narzędzi wspomagających projektowanie i programowanie, posługiwaniu się językiem C,C++ (aczkolwiek używany jest również język Clipper i pliki typu dbf nie bardzo nadające się do eksploatacji w charakterze dużych baz danych). W sumie przygotowanie programistów jest dobre, gdy w aspekcie przewidy-

wanego dalszego rozwoju systemów od strony technologii informatycznej oraz wzbogacenia funkcjonalnego występującego w tzw. międzynarodowych (world-wide) systemach bankowych. Potrzebne będzie jednak uzupełnienie wiedzy programistów o tematykę, m.in. zarządzania skarbowością, obsługi operacji zagranicznych, papierów wartościowych, itp.

To przejście na etap zaawansowanej bankowości komercyjnej i zarządzania finansami banku oraz konieczność wbudowania funkcji zarządzania limitami i ryzykiem w przekroju dealerów, klientów, walut, branż, krajów, itp. skomplikuje architekturę aplikacyjną systemów i technologię przetwarzania.

Nastąpi zwiększenie roli czasu rzeczywistego m. in. do uzyskiwania pozycji klienta, przepływu pieniężnego (aktualnego i przewidywanego) oraz ewentualnie do prognozowania bilansu księgowego. Prawdopodobnie nastąpi przejście w kierunku scentralizowanych rozwiązań na mini (lub midi) komputerach usadowionych w centrali banku.

Firmy software'owe, które pierwsze udoskonalą swoje systemy i zapewnią sobie niezbędny dopływ wiedzy bankowej, będą mieć stabilniejszą pozycję w nadchodzących latach, szczególnie w obliczu tzw. otwarcia od 1997 r krajowego rynku finansowego dla banków zachodnich.

g) Zdecydowana przewaga zdecentralizowanych rozwiązań

Dużo jest zdecentralizowanych baz danych i zdecentralizowanego oprogramowania.

h) Działania rozwojowe w kierunku pełnej wielowalutowości oraz architektury klient/serwer

i) Typową platformą są komputery osobiste przy czym w przypadku stosowania systemu UNIX w grę wchodzą minikomputery typu HP9000 czy Digital ALPHA. W związku z przewidywaną konsolidacją banków i brakiem odpowiedniej przepustowości komputerów osobistych, należy liczyć się z przesuwaniem ciężaru przetwarzania na minikomputery, pozostawiając „pecetom" rolę stacji roboczych i emulowanych terminali. Powstaną również warunki do stosowania zaawansowanych systemów zarządzania bazami danych, lecz ich wprowadzenie nie może prowadzić do „przebudowania" architektury systemu i technologii przetwarzania.

j) Występuje stosunkowo słabe zabezpieczenie systemów (istnieją tutaj pozytywne wyjątki w szczególności w zakresie homebankingu) w zakresie dostępu, tworzenia śladów audytowych oraz przywracania stanu systemu po awarii lub zabezpieczenia przez centra zapasowe (disaster centre). W szczególności problematyka śladów audytowych jest często nie doceniana przez krajowych projektantów i programistów.

k) Istnieją powiązania z otoczeniem zewnętrznym.

Powszechnie stosowane jest łącze do KIR/Elixir(czasem do sieci SWIFT oraz bankomatów).

Pożądane i przewidywane zmiany w systemach krajowych

Pierwszy etap budowy systemów bankowych, zorientowany na obsługę detalicznej działalności oddziałów, został w kraju już zakończony i wyniki jego uznać należy za zadowalające, gdyż nastąpiło to w stosunkowo krótkim czasie (zważywszy na 10-15 letnie opóźnienie w stosunku do Zachodu - jakie istniało w 1990 roku- a więc w momencie rozpoczynania działalności firm polskich) , natomiast jakość rozwiązań w większości systemów jest zadowalająca.

Tendencja konsolidowania banków oraz zwiększająca się rola banków zagranicznych i komercyjnych mogą spowodować rezygnację banków z dotychczasowych systemów, co może postawić w trudnej sytuacji krajowych wytwórców oprogramowania bankowego.

Wzajemna konkurencja (większość systemów posiada zbliżony zakres funkcjonalny, ograniczający się do bankowości detalicznej) i konsolidacja banków spowodują zapewne wzrost wymagań, a w rezultacie pojawi się zapotrzebowanie na systemy „inteligentne", które nie tylko pozwolą sprawnie obsłużyć klientów przy okienku lecz także poradzą, gdzie i jak zainwestować pieniądze przy najmniejszym stopniu ryzyka (również na rynkach zagranicznych).

Warunki prenumeraty informatyki na 1998 rok

Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując opłaty na adres:

Wydawnictwo SIGMA-NOT Spółka z o.o., Zakład Kolportażu, 00-716 Warszawa, skr. pocztowa 1004
konto: PBK S.A. III O/Warszawa
nr 11101024-1573-2720-3-28

Prenumerata ulgowa dotyczy członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT, członków PTI, uczniów szkół średnich oraz studentów szkół wyższych.
Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT, PTI lub szkoły.

Cennik rocznej prenumeraty: normalna – 72,00 zł, ulgowa – 36,00 zł
Cena 1 egzemplarza wynosi 6,00 zł, ulgowa 3,00 zł.

Wszystkie pytania i wątpliwości
prosimy kierować do Zakładu Kolportażu:
ul. Bartycka 20, paw. B, 00-950 Warszawa, skr. poczt. 1004,
tel./fax: 40-35-89, tel. 40-30-86, centr. 40-00-21 w. 295

Prenumeratę przyjmujemy również w sieci internet:
www.pol.pl./sigma_not *oraz* **e-mail: kolpor.sigma@pol.pl**

Dystrybucja:

Clico Centrum Oprogramowania sp. z o.o.
Al. 3 Maja 7, 30-063 Kraków tel.(012)-32-51-66, 34-34-22 w.146
Fax: 12 32-36-98; Email:support@clico.krakow.pl
WWW.clico.krakow.pl FTP.clico.krakow.pl

CLICO

Nie ma znaczenia czy jesteś duży czy mały.
Ważne jest, jak się rozwijasz.

Każda firma, duża czy mała chce osiągnąć sukces. Sukces i ciągły wzrost to nie tylko właściwy sposób myślenia i planowania w przedsiębiorstwie, zarządzania ludźmi i środkami materialnymi, lecz również umiejętność przystosowywania się do zachodzących w otoczeniu zmian.

Sukces to także gwarancja szybkiej wymiany informacji między wszystkimi pracownikami biorącymi udział w różnych etapach procesu technologicznego.

W jego osiąganiu pomagają narzędzia, takie jak system SAP R/3.

System R/3 jest zbiorem aplikacji integrujących procesy gospodarcze w całym przedsiębiorstwie, począwszy od zarządzania finansami, poprzez rozliczanie kosztów, planowanie i organizację produkcji, sprzedaży, gospodarkę materiałową, remonty, planowanie inwestycji do zarządzania projektami i kadrami firmy. System R/3 pozwala działać szybko i efektywnie. Jest dostępny w polskiej wersji językowej i jest dostosowany do polskich regulacji prawnych. Wdrażają go w Polsce wykwalifikowani konsultanci.

R/3 jest systemem standardowym, wykorzystującym najnowocześniejsze technologie informatyczne, niezależnym od platformy sprzętowej i systemowej oraz mającym stabilną, modułową konstrukcję. Daje to gwarancję niezawodności i bezpieczeństwa inwestycyjnego, a jednocześnie sprawia, że okres wdrożenia jest relatywnie krótki.

Z drugiej strony, dzięki szerokim możliwościom parametryzacji i adaptacji, można dostosować go do indywidualnej struktury organizacyjnej każdego przedsiębiorstwa i instytucji.

SAP R/3 pomaga prawidłowo zarządzać przedsiębiorstwem. Teraz i w przyszłości. Znajduje to potwierdzenie już w ponad 11000 instalacji systemu R/3 na całym świecie, niezależnie od wielkości i charakteru działalności firmy.

Jeżeli macie Państwo pytania dotyczące systemu R/3, chętnie odpowiemy na nie i prześlemy materiały informacyjne.

Nasz adres:
SAP Polska Sp. z o.o. Mokotów Business Park
02-672 Warszawa, ul. Domaniewska 41
tel. (+48 22) 606 06 06; fax (+48 22) 606 06 07

SAP®
A Better Return On Information.

Osadzanie obiektów multimedialnych w bazach danych

Piotr Porwik, Wojciech Jurczyk, Adam Trojan

Współcześnie rozwijane technologie informatyczne umożliwiają gromadzenie kolekcji danych zawierających informacje multimedialne w postaci obrazów, dźwięków lub dokumentów. Wiele dostępnych na naszym rynku języków projektowania baz danych (Magic, Delphi, FoxPro, itp.) posiada wbudowane mechanizmy rejestracji i aktualizacji takich typów danych.

Filozofia implementowania struktur multimedialnych w tworzonych aplikacjach jest wszędzie bardzo podobna bez względu na platformę oprogramowania. Istniejące różnice wynikają raczej z właściwości wybranego do tego celu narzędzia będącego wypadkową jego wad i zalet. Nie ma jednak czytelnych reguł, za pomocą których można przygotować strukturę i zaproponować obsługę bazy dla zastosowań multimedialnych. W zasadzie zawsze istnieją możliwości rejestracji zdarzeń multimedialnych w bazie, ale obsługa takich obiektów nie jest w wielu przypadkach przejrzyście udokumentowana, a nawet możliwa.

W artykule przedstawiono metody wykorzystywania obiektów multimedialnych w bazach danych, zwracając uwagę na pewne uniwersalne mechanizmy metodologiczne potrzebne przy projektowaniu takich aplikacji.

Struktura bazy i definicje pól, język programowania

Struktura każdej bazy uzyskuje pewne predefiniowane właściwości, uzyskiwane na etapie tworzenia projektu. Cechy te nie są widoczne podczas projektowania, ale można je modyfikować w przypadku bardziej wyrafinowanych metod kształtowania bazy. W ten sposób można dla poszczególnych tabel bazy modyfikować prawa dostępu, hasła uprawnień, identyfikować uprawnionych użytkowników bazy, itp. Znajomość tych atrybutów nie jest jednak potrzebna do zrozumienia zasad budowy aplikacji multimedialnej.

Opisane cechy oraz inne atrybuty budowanej aplikacji nadawane mogą być za pomocą zintegrowanego środowiska programowego :

- **narzędzi do tworzenia aplikacji w środowisku Windows.** Umożliwiają one uruchamianie aplikacji jednostanowiskowych, a także projektowanie i wykonywanie instalacji sieciowych. W obydwu przypadkach wykorzystywana jest technologia klient-serwer oraz udogodnienia języków 4GL. Praktycznie wszystkie obecne na rynku pakiety programowe dysponują własnymi językami opisu rzeczywistości (np. SQLWindows Aplication Language - SAL, oferowany przez firmę *Gupta)*, za pomocą których może być zbudowana prawie cała aplikacja. Język ten jest w zasadzie zbiorem sparametryzowanych funkcji (często obiektowych), które dostępne są już na etapie zapisu programu z poziomu specjalizowanego edytora programów źródłowych. Dodatkowo dostępne są podstawowe instrukcje znane z klasycznych języków programowania: break, call, else, if, loop, return, set, while, when, case, itp., które nie wymagają specjalnego omawiania.

- **serwerów relacyjnych baz danych** Serwer baz danych (lokalny lub sieciowy) przechowuje poszczególne bazy w plikach. Przeważnie na etapie projektowania bazy serwer tworzy dla niej katalog o tej samej nazwie co baza. Baza może zawierać jedną lub wiele tabel, mających swe odrębne nazwy i struktury. Podobnie jak w klasycznej teorii relacyjnych baz danych tabela jest zbiorem rekordów, z których każdy podzielony jest na pola. Dla każdego pola zdefiniować można atrybuty, które mogą przyjmować wartości z określonego zbioru (dziedziny). Nazwy pól przyjmowane są według powszechnie uznanego standardu i nie wymagają dodatkowych objaśnień.

Znakowe (String):
- *CHAR*
- *VARCHAR*
- *LONG VARCHAR*

Liczbowe (Number):
- *NUMBER*
- *DECIMAL*
- *INTEGER*
- *SMALINT*
- *DOUBLE PRECISION*
- *REAL*
- *FLOAT*

Daty i czasu (Date/Time):
- *DATETIME*
- *DATE*
- *TIME*

Zmienne używane w aplikacjach multimedialnych nazywane (identyfikowane) są według powszechnie stosowanego standardu i mogą być następujących typów: *Boolean, Date/Time,*

Number, Long String, String, File Handle, Sql Handle, Window Handle.

Jedynym typem, w którym można przechowywać dane większe niż 254 bajtów, jest typ *LONG VARCHAR* (dla pól tabeli bazy danych) i *Long string* dla zmiennych występujących w aplikacji. Z tych powodów jest on przeznaczony dla danych multimedialnych. Teoretycznie tego typu danych można używać także dla przechowywania dowolnej wartości typu znakowego. Jest to jednak nieekonomiczne. Fizycznie zmienne typu *Long string* przechowują informacje dodatkowe, które generowane są już na etapie uruchamiania aplikacji i oprócz zasadniczej informacji medialnej (grafika, dźwięk, tekst) mogą zawierać dodatkowe deskryptory przechowujące informacje o pochodzeniu obiektu. Skuteczna manipulacja danymi tego typu wymaga więc użycia specjalizowanych funkcji języka (patrz punkt a).

Podczas projektowania struktury rekordu należy w standardowej bazie zdefiniować typ pola. Każde pole charakteryzuje się określoną liczbą bajtów, jaką zajmie w rekordzie. W przypadku pola typu *LONG VARCHAR* informacja o jego szerokości jest zbędna, gdyż po prostu nie jest jeszcze znana na tym etapie przygotowania bazy. W polu tym w momencie osadzania (przez wbudowanie lub dołączanie) obiektu przechowywany jest identyfikator dokumentu źródłowego i jego treść wraz z ewentualną nazwą aplikacji tworzącej format dokumentu źródłowego. Ogólna zasada umieszczania i wyświetlania obiektu multimedialnego w bazie przedstawiona została na rysunku.

Manipulowanie danymi multimedialnymi

Do wyświetlania ilustracji multimedialnych służy obiekt typu *PICTURE* (rys.1), który może być częścią okna aplikacji lub okna dialogowego. Źródłem danych dla tego obiektu mogą być:

- wspólny obszar pamięci Windows: schowek (Clipboard);
- serwery OLE;
- plik lub baza danych.

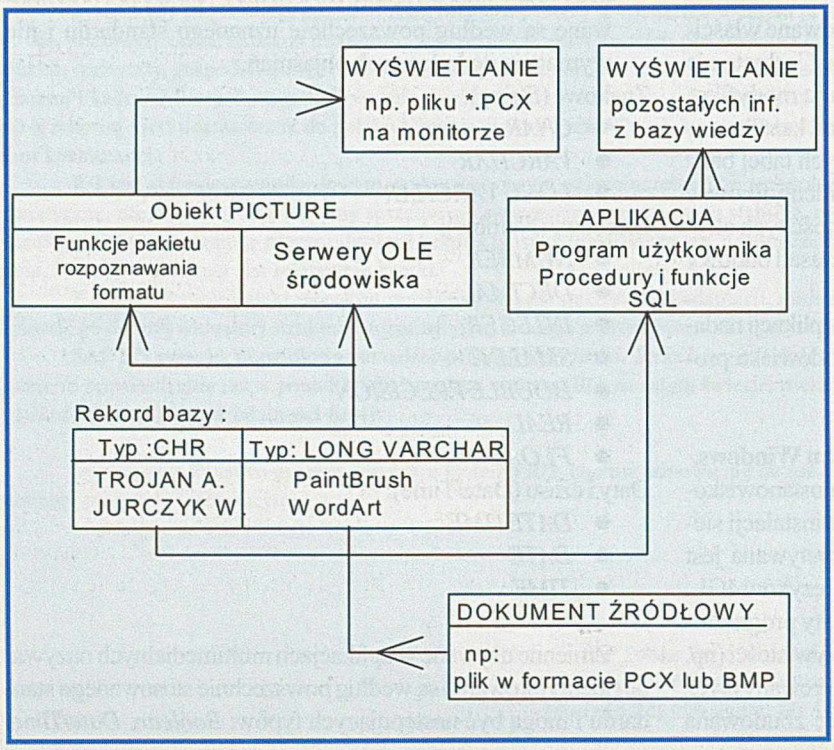
Dane umieszczane w obiekcie *PICTURE* mogą być zapisywane lub odczytywane za pomocą zestawu funkcji dostępnych z poziomu używanego przez programistę oprogramowania. Mając zarejestrowane obiekty multimedialne w bazie, można prezentować je na ekranie monitora poprzez mechanizmy dostępne bezpośrednio z poziomu aplikacji. Oznacza to, że tylko wewnętrzne mechanizmy danego pakietu odpowiedzialne są za wizualizację obiektu graficznego. Przeważnie można prezentować ilustracje zapisane za pomocą przynajmniej podstawowych formatów opisu grafiki (tabela).

Typ pliku	Opis
Tag Image File Format	TIFF
Paintbrush	PCX
Graphics Interchange Format	GIF
Device-Independent Bitmap	BMP, DIB
Icon File	ICO
Windows MetaFile	WMF
No Graphic File	DOSFILE

Jeśli nie jest znany proces źródłowy, za pomocą którego utworzona została ilustracja multimedialna, użytkownik powiadamiany jest tylko o typie rozpoznanego obiektu i sposobie jego wyświetlania : BMP, GIF, WMF, itp.

W przypadku kiedy znana jest aplikacja źródłowa, za pomocą której utworzono obiekt multimedialny, wiadomości przekazywane użytkownikowi opisują źródło generujące, np. Package Object, Sound Object, Paintbrush Picture Object, itp.

Wiele środowisk programowych nie dostarcza narzędzi do samodzielnego tworzenia grafiki. Tego typu obiekty należy wówczas tworzyć w innych programach. Jednak po zastosowaniu odpowiednich zabiegów programowych można utworzyć na ekranie obiekt graficzny bezpośrednio z własnej aplikacji. Realizację tego zapewniają funkcje API (ang. Application Program Interface) zawarte w bibliotekach gdi.exe oraz user.exe środowiska Windows.



Tak wyprodukowane obrazy graficzne nie mogą jednak zostać umieszczony na stałe w bazie danych, gdyż nie można zawartości okna identyfikowanego unikalnym identyfikatorem przypisać żadnej zmiennej (tutaj - typu *Long string*). Wszystkie zapisy i odczyty baz danych w językach 4GL odbywają się jak wiadomo w standardzie *SQL* (select, insert, update, delete), gdzie wymagana jest dostępność zmiennych aplikacji, dzięki którym następuje wybór, wpisywanie, modyfikacja lub kasowanie odpowiednich pól tabeli bazy. Omawiany przykład jest więc bardziej opisem dydaktycznym niż sposobem na wykonanie prezentacji graficznej.

Wykorzystując dostępne mechanizmy stosowanych przez programistę pakietów oraz stosując znane techniki OLE środowiska Win-

dows, można umieszczać zadowalająco obiekty multimedialne w bazach danych na kilka sposobów:

1. Metoda bezpośredniego zapisu informacji (niekoniecznie multimedialnej) do odpowiedniego pola bazy danych przez zmienną aplikacji typu *Long string*;
2. Metoda zapisu informacji do bazy danych poprzez obiekt *PICTURE.;*
3. Zapis danych multimedialnych poprzez obiekt *PICTURE* i dostępne dla aplikacji serwery OLE (ang. Object Linking and Embedding).

Każdy z wymienionych sposobów ma swoje zalety i wady, które omówione będą w dalszej części artykułu. Nieprawidłowa metoda projektowania aplikacji medialnej (np. błędny dobór niektórych predefiniowanych stałych) może spowodować niewłaściwe wyświetlanie informacji graficznej przy równoczesnym braku komunikatów o błędach, pomimo że projektant aplikacji przewidział miejsca indykowania niewłaściwych stanów.

Zapisywane w bazie danych ilustracje graficzne - jeśli tylko to możliwe- uzupełniane są unikalnym opisem formatującym. Opis taki umożliwia w przypadkach późniejszych manipulacji jednoznaczną identyfikację i klasyfikację odczytywanych danych multimedialnych.

Metoda 1. Bezpośredni zapis informacji do pola bazy danych

W takim przypadku wykorzystuje się wyłącznie funkcje operujące na fizycznym pliku, za pomocą których odczytywane są odpowiednie dane, szacowana jest ich wielkość, a następnie poleceniem *SQL insert* wykonywany jest zapis odczytanej informacji bezpośrednio do tabel bazy danych.

Opisana metoda pozwala wprawdzie umieścić w bazie danych dowolny plik (także nie będący reprezentantem obiektu graficznego !), ale w przypadku wyświetlania takich danych na ekranie monitora mogą wystąpić błędy odtwarzania. Jak powiedziano powyżej, każda informacja multimedialna przed jej prezentacją na ekranie podlega wstępnemu przetworzeniu celem ustalenia jej formatu (typu). Zapis informacji powyższą metodą (bezpośrednio do bazy danych z pliku dyskowego) wykonany zostaje bez analizy danych.

Metoda 2. Zapis informacji do bazy danych poprzez obiekt PICTURE

Zapis danych z wykorzystaniem tej metody umożliwia kontrolę formatu odczytywanych i zapisywanych danych. Tym sposobem integralne mechanizmy aplikacji nadają przetwarzanej informacji przed jej rejestracją w bazie specjalne (dodatkowe) własne opisy, które następnie wykorzystywane są do wewnętrznej identyfikacji formatu graficznego. Umożliwia to w konsekwencji pobieranie danych multimedialnych z bazy i bezpośrednią ich prezentację z poziomu aplikacji bez korzystania z usług zewnętrznych serwerów OLE. W tym celu podczas pobierania danych z bazy ładowane są równocześnie w tle wybrane biblioteki DLL, których funkcje odpowiedzialne są za konwersję i prezentację pliku graficznego. Omawiana metoda pozwala więc na zapisywanie i odtwarzanie plików graficznych na przykład w formatach wymienionych w tabeli 1. Zmiany obrazu bazy danych odbywają się za pomocą odpowiedniego polecenia *SQL* (insert lub update).

W niektórych pakietach (np. *Gupta*) obydwie opisane metody nie pozwalają w sposób naturalny odtwarzać ilustracji muzycznych, gdyż zarejestrowany plik (np. *.wav) podczas zapisu do bazy danych nie jest kojarzony z żadnym programem świadczącym usługi nagrywania/odtwarzania dźwięków. W takich przypadkach należy wywoływać funkcje API Windows, np. *SndPlaySound()* z biblioteki mmsystem.dll, której parametrem jest nazwa pliku podlegającemu odtworzeniu. Inne pakiety (np. *Delphi*) posiadają wbudowane mechanizmy do odtwarzania dźwięku rozpoznające formaty zapisu i na tej podstawie uruchamiają własne wewnętrzne narzędzia do odtwarzania/nagrywania dźwięku.

Metoda 3. Zapis informacji multimedialnej z wykorzystaniem serwerów OLE

Technika OLE jest metodą transferu i łączenia danych pomiędzy różnymi aplikacjami środowiska systemu operacyjnego Windows. W metodzie tej wykorzystuje się mechanizmy związane z osadzaniem obiektów multimedialnych w aplikacjach. Wyróżnić można dwa typy aplikacji OLE :

- Serwery (ang. servers): aplikacje odpowiedzialne za tworzenie oraz edycję obiektów;
- Klienci (ang. clients): aplikacje, w których lokujemy obiekty.

Na przykład, jeśli chcemy umieścić za pomocą uruchomionej aplikacji ilustrację tworzoną przez edytor graficzny Paintbrush, to aplikacja ta jest aplikacją-klientem, edytor Paintbrush jest aplikacją-serwerem, a ilustracja Paintbrush jest obiektem. Technika OLE traktuje wszystkie dane jako obiekty, którymi mogą być tekst, grafika lub dźwięk.

Istnieją dwa typy obiektów OLE :

– **Obiekty wbudowane (ang. *embedded objects*).**
W tego typu obiekcie wszystkie informacje potrzebne do jego utworzenia podlegają enkapsulacji i zawarte są bezpośrednio w pliku utworzonym przez serwer OLE. Dane te zawierają nazwę serwera, który utworzył obiekt, wewnętrzne dane obiektu oraz sposób prezentacji obiektu w aplikacji klienta. Edycja obiektu wbudowanego polega na wybraniu odpowiedniego pliku źródłowego z menu i nie powoduje zmian w tym materiale, chociaż w takim przypadku automatycznie użyty zostaje edytor, za pomocą którego powstała ilustracja źródłowa. Podobnie, bezpośrednia edycja materiału źródłowego nie spowoduje zmian w obiekcie wbudowanym. Stosując taką metodę umieszczania obiektów multimedialnych, decydujemy, że istnieją dwie kopie materiału źródłowego, z których jedna jest wbudowana bezpośrednio w aplikację docelową, a drugą stanowi plik utworzony przez serwer OLE. Po operacji wbudowania plik ten można bez przeszkód z dysku usunąć i dalsze modyfikacje przeprowadzać z poziomu aplikacji docelowej. Obiekty wbudowane są obiektami statycznymi i umieszczane są w aplikacji docelowej metodą wklejania (ang. paste).

– **Obiekty powiązane (ang. *linked objects*).**
Metoda ta charakteryzuje się dynamicznym powiązaniem informacji - zmiany dokonywane w obiekcie za pomocą pro-

gramu źródłowego (serwer OLE) zostaną automatycznie naniesione na ilustrację znajdującą się w aplikacji docelowej. Niektóre definicje takich technik OLE [1,4] podają, że w aplikacji docelowej powinny się znaleźć wyłącznie referencje zawierające nazwę serwera, który utworzył obiekt oraz nazwa pliku danych źródłowych wraz ze ścieżką dostępu. Jak wynika z obserwacji większość aplikacji przechowuje jednak dodatkowo (w postaci drugiej kopii) wewnętrzne dane pliku źródłowego. W zależności od indywidualnych rozwiązań programowych utrata (uszkodzenie) pliku źródłowego może powodować różne skutki w aplikacji docelowej. Może to być np. utrata zdolności edycji obiektu, zerwanie połączenia z serwerem OLE, zmiana cech obiektu z powiązanego na wbudowany lub wystąpienie błędu krytycznego w programie.

Prawidłowe postępowanie prowadzi jednak zawsze do tego, że z jedną kopią oryginalnego materiału źródłowego można wiązać wiele innych aplikacji docelowych. We wszystkich tak powiązanych aplikacjach dokonane zmiany w materiale oryginalnym będą uwidaczniane w aplikacjach docelowych. Druga kopia zawarta jest bezpośrednio w aplikacji docelowej. Modyfikacje dokonywane z tego poziomu mogą skutkować różnymi efektami opisanymi powyżej.

W aplikacjach multimedialnych przedstawione techniki OLE rozpatrywać można w trzech odmianach :

Najpierw należy zawsze wskazać lub utworzyć plik podlegający osadzeniu, wykonać opisane poniżej czynności oraz zakończyć sesję zapisu danych wydaniem odpowiedniego polecenia *SQL*.

A. Pierwsza opcja polega na wykorzystaniu funkcji, które wstawiają do obiektu *PICTURE* zawartość wskazanego pliku dyskowego. Aplikacja kojarzy (wiąże) wstawiany plik z Pakowarką (Object Package). Jak wiadomo pakowarka jest narzędziem, za pomocą którego możemy prezentować dokumenty w formie spakowanej - pod postacią wskazanego piktogramu (ikony). Tym sposobem na ekranie monitora zostaje wyświetlona ikona - symbol aplikacji (serwera OLE) odpowiedzialnej za utworzenie pliku źródłowego wraz z nazwą pliku. Stosując tę metodę, nie można dowolnie wybierać serwerów, które byłyby najodpowiedniejsze do naszych celów, gdyż serwer taki zostaje przydzielony automatycznie przez system Windows na podstawie analizy dokumentu źródłowego, który chcemy spakować. Jest to także jedna z metod prezentacji dźwięku. Do odtworzenia nagrania wykorzystywane są oczywiście istniejące zasoby Windows (np. sound recorder). Zaprezentowany mechanizm jest typowy dla osadzania obiektów w systemie Windows metodą wbudowania.

B. Druga opcja polega na skojarzeniu wskazanego samodzielnie serwera OLE z obiektem *PICTURE*. Za pomocą specjalizowanych funkcji pakietu można jawnie pobierać informacje na temat zarejestrowanych w systemie serwerów, jak i świadczonych przez nie usług. Zrealizowanie usługi wymaga więc otwarcia okna dialogowego, w którym musimy wskazać plik, w którym zapisana jest informacja źródłowa (multimedialna) i w dalszej kolejności wybrać odpowiedni z naszego punktu widzenia serwer OLE oraz jego usługę. W bazie danych umieszczana jest kompletna informacja multimedialna wraz z referencjami odpowiedniego serwera.

C. Trzecia metoda polega na takim umieszczeniu ilustracji w aplikacji użytkownika, aby zapewnić dynamiczne powiązanie serwera OLE i obiektów multimedialnych osadzonych w aplikacji docelowej. Oznacza to tym samym, że w bazie danych przechowywana powinna być jedynie nazwa obiektu (pliku), który chcemy prezentować na ekranie oraz referencje serwera OLE odpowiedzialnego za wykonanie tego zadania. Rozwiązanie takie (z pewnymi ograniczeniami dotyczącymi aktualizacji danych) jest możliwe, jednak z punktu widzenia zasad dobrego programowania jest najgorszym z możliwych, gdyż tym sposobem tracimy zupełnie panowanie nad integralnością bazy danych. Baza danych nie zawiera wówczas kolekcji danych multimedialnych, a jest jedynie miejscem przechowywania formuł sterujących wyświetlaniem obiektów graficznych. Wszystkie zasadnicze ilustracje multimedialne znajdują się poza samą bazą danych, co jest nie do przyjęcia.

Osadzanie obiektów multimedialnych w bazach danych nie jest sprawą oczywistą i wymaga wcześniejszej, dokładnej analizy potrzeb użytkownika aplikacji. Różne, nie zawsze jednoznacznie opisywane techniki manipulacji danymi powodują, że projektowanie aplikacji multimedialnej jest szczególnie wrażliwe na błędy programisty. Dodatkowo jak wskazuje praktyka autorów, w różnych pakietach programowych wykorzystywanie ich właściwości multimedialnych jest związane ze specyficzną filozofią producenta oprogramowania, co powoduje, że techniki projektowe zastosowane w jednej aplikacji mogą prowadzić do nieoczekiwanych efektów w aplikacji utworzonej za pomocą innego pakietu programowego. Niestety nie istnieje jeszcze powszechnie obowiązujący standard dotyczący sposobów wykorzystywania multimediów w bazach danych oraz wewnętrznych, aplikacyjnych formatów związanych z zapisem i odczytem takich danych. Techniki OLE również nie są jednolicie wykorzystywane, ich interpretacja w pakiecie (niezależna od projektanta systemu) zmusza do szczególnej uwagi przy projektowaniu aplikacji. Podstawowy opis technik OLE umożliwia lepsze zrozumienie problemu.

Należy podkreślić, że parametry posiadanego sprzętu komputerowego również znacząco wpływają na ostateczne walory użytkowe projektowanej aplikacji [3].

W większości pakietów programowych opisywane sposoby i techniki manipulacji obiektami multimedialnymi są dostępne z poziomu tworzonego oprogramowania poprzez wywoływanie odpowiednich funkcji lub procedur. Interpretacja ich jest jednak niejednolita - różna u różnych producentów oprogramowania.

Literatura

[1] Gupta Corporation. SQLWindows Documentation. Function Reference,1995;
[2] Oracle Corporation „Poradnik administratora bazy danych ORACLE RDBMS",1990;
[3] P. Porwik. „Multimedialne bazy wiedzy-rekomendacje", *Informatyka*, nr10,1993;
[4] Microsoft Windows Documentation. Manual Reference;
[5] Wacławek „Windows od kuchni", Oficyna Wydawnicza Help, Warszawa, 1993.

Piotr Porwik, Wojciech Jurczyk*, Adam Trojan*
są pracownikami Instytutu Informatyki Stosowanej Uniwersytetu Śląskiego
*Koło Naukowe Elektroników i Informatyków UŚl.

SYSTEMY CZASU RZECZYWISTEGO

REDAGUJE ZESPÓŁ: dr inż. Zbigniew FRYŹLEWICZ (Politechnika Wrocławska), prof. dr hab. inż. Janusz GÓRSKI (Politechnika Gdańska), prof. dr hab. inż. Zbigniew HUZAR (Politechnika Wrocławska), dr hab.inż. Krzysztof SACHA (Politechnika Warszawska), prof. dr hab. inż. Tomasz SZMUC (Akademia Górniczo-Hutnicza), dr inż. Zbigniew Zieliński (Wojskowa Akademia Techniczna), mgr inż. Zdzisław ŻURAKOWSKI (Instytut Automatyki Systemów Energetycznych we Wrocławiu) - przewodniczący zespołu.

ROK IV • LISTOPAD • 1997 • 32

Inspekcje produktów informatycznych

Marek Jurkowlaniec, BjØrn Runge

Artykuł jest poświęcony inspekcji jako jednej z technik stosowanych w procesie zapewnienia jakości produktów informatycznych. Zawiera wprowadzenie, omówienie poszczególnych faz oraz ról uczestników inspekcji. Przedstawiono takie zagadnienia, jak: dokumentacja, formularze oraz metryki związane z przebiegiem inspekcji. Artykuł podsumowuje dyskusja na temat kosztów i potencjalnych korzyści płynących z stosowania tej techniki, jak również uwarunkowań sukcesu jej wprowadzenia.

Gwarancja jakości odgrywa zasadniczą rolę w procesie wytwarzania oprogramowania i powinna towarzyszyć jego każdej fazie i aktywności. Wbudowanie mechanizmów projakościowych do procesu wytwarzania umożliwia wczesne wykrywanie i rozwiązywanie problemów. Zmniejsza to później koszt ich usuwania: koszt naprawy błędu wprowadzonego na etapie analizy wymagań, zidentyfikowanego i usuniętego w trakcie projektowania systemu jest 10-krotnie większy niż koszt jego usunięcia na etapie wymagań. Usunięcie takiego błędu w trakcie implementacji systemu może kosztować już 100 razy więcej, a w trakcie eksploatacji - nawet 1000 razy [3]. Wyniki analiz dotyczących jakości oprogramowania wskazują, że większość (ok. 60-70%) błędów zostaje wprowadzona w fazach analizy i projektowania, a jedynie 30-40% - w trakcie kodowania. Jeśli uwzględnimy, że testowanie umożliwia wykrycie jedynie ok. 50% błędów (i to w większości tych powstałych w procesie kodowania), okaże się, że większość nieprawidłowości pozostaje w produkcie dostarczonym odbiorcy i ujawni się dopiero w trakcie eksploatacji. Dane te prowadzą do prostego wniosku: należy wprowadzić mechanizmy, które umożliwią eliminację nieprawidłowości w możliwie najkrótszym czasie po ich wprowadzeniu. Dotyczy to każdego etapu wytwarzania oprogramowania.

Inżynieria oprogramowania jest dziedziną bardzo młodą w porównaniu z „tradycyjnymi" dziedzinami inżynierii. Ze względu na swą specyfikę (gwałtowny rozwój technologii informatycznej, brak tradycji, etos programisty - geniusza, traktowanie programowania jako swoistej sztuki) nie dopracowano jeszcze w pełni dojrzałych modeli procesów wytwarzania. Na powstałą sytuację ma też wpływ dominująca rola kodu źródłowego i gotowego produktu (czyli działającego programu) kosztem jego wcześniejszych reprezentacji (wymagań, projektu). Tendencja ta znajduje wyraz w nieczęstym poruszaniu problematyki jakościowej przy produkcji oprogramowania. Dla wielu fachowców z tej branży gwarancja jakości sprowadza się wyłącznie do testowania działającego programu.

Ostatnie lata przyniosły znaczną poprawę w tym zakresie [HUPM89]. Współczesne modele procesów wytwarzania oraz międzynarodowe standardy inżynierii oprogramowania zakładają obecność zadań zapewnienia jakości w całym procesie wytwarzania oprogramowania. Zależnie od fazy procesu, jak i rodzaju dokumentacji poddanej zabiegom projakościowym, stosowane są różnego rodzaju metody i procedury.

Przedmiotem zainteresowania niniejszego artykułu jest *inspekcja* - zespołowa metoda wykrywania defektów w produktach procesu wytwarzania oprogramowania. Ze względu na jej wysoką efektywność i uniwersalność jest to jedna z najważniejszych metod zagwarantowania jakości.

Technika inspekcji została zaproponowana przez Michaela E. Fagana, specjalistę do spraw jakości w laboratoriach IBM w Kingston, NY. W latach 1972-74 był on menedżerem projektu translacji oprogramowania IBM z języka Asembler na PLS. Zdecydował, że w projekcie tym zostaną

zastosowane statystyczne metody jakościowe, znane z innych gałęzi przemysłu. Wyniki opublikował w znanym artykule [4]. Technika (nazwana później „Inspekcją Fagana") została zastosowana w IBM, początkowo do kodu i pseudokodu, później - do wszelkich form produktów procesu wytwarzania oprogramowania. Metoda przyniosła nadspodziewanie dobre wyniki w zakresie poprawy zarówno jakości produktu, jak i efektywności procesu. Za swe osiągnięcia Fagan otrzymał od IBM wysoką nagrodę.

Inspekcja jest metodą formalną w sensie organizacyjnym. Standard IEEE [1] definiuje inspekcję jako: „...formalną technikę oceny, w ramach której wymagania na oprogramowanie, projekt lub kod są poddane badaniu przez osobę lub grupę osób innych niż ich autor w celu wykrycia błędów, odstępstw od standardów wytwarzania i innych problemów...". Spośród celów inspekcji wymienić można:

- wykrywanie i identyfikację defektów w produktach procesu wytwarzania oprogramowania (rozumianych jako specyfikacja wymagań, kod, dokumentacja, specyfikacja testów i inne);
- usunięcie tych defektów;
- formalne potwierdzenie jakości produktu;
- doskonalenie procesu wytwarzania oprogramowania na podstawie wyników analizy przyczyn znalezionych defektów;
- doskonalenie samego procesu inspekcji.

Inspekcja to złożone przedsięwzięcie organizacyjne obejmujące wiele kroków prowadzonych na podstawie zdefiniowanych procedur i dokumentów pomocniczych z udziałem zespołu kompetentnych specjalistów o zdefiniowanych rolach. Najczęściej jest przeprowadzana w końcowych fazach etapów procesu wytwarzania, podczas których powstają kluczowe elementy oprogramowania, takie jak specyfikacja wymagań czy kod źródłowy.

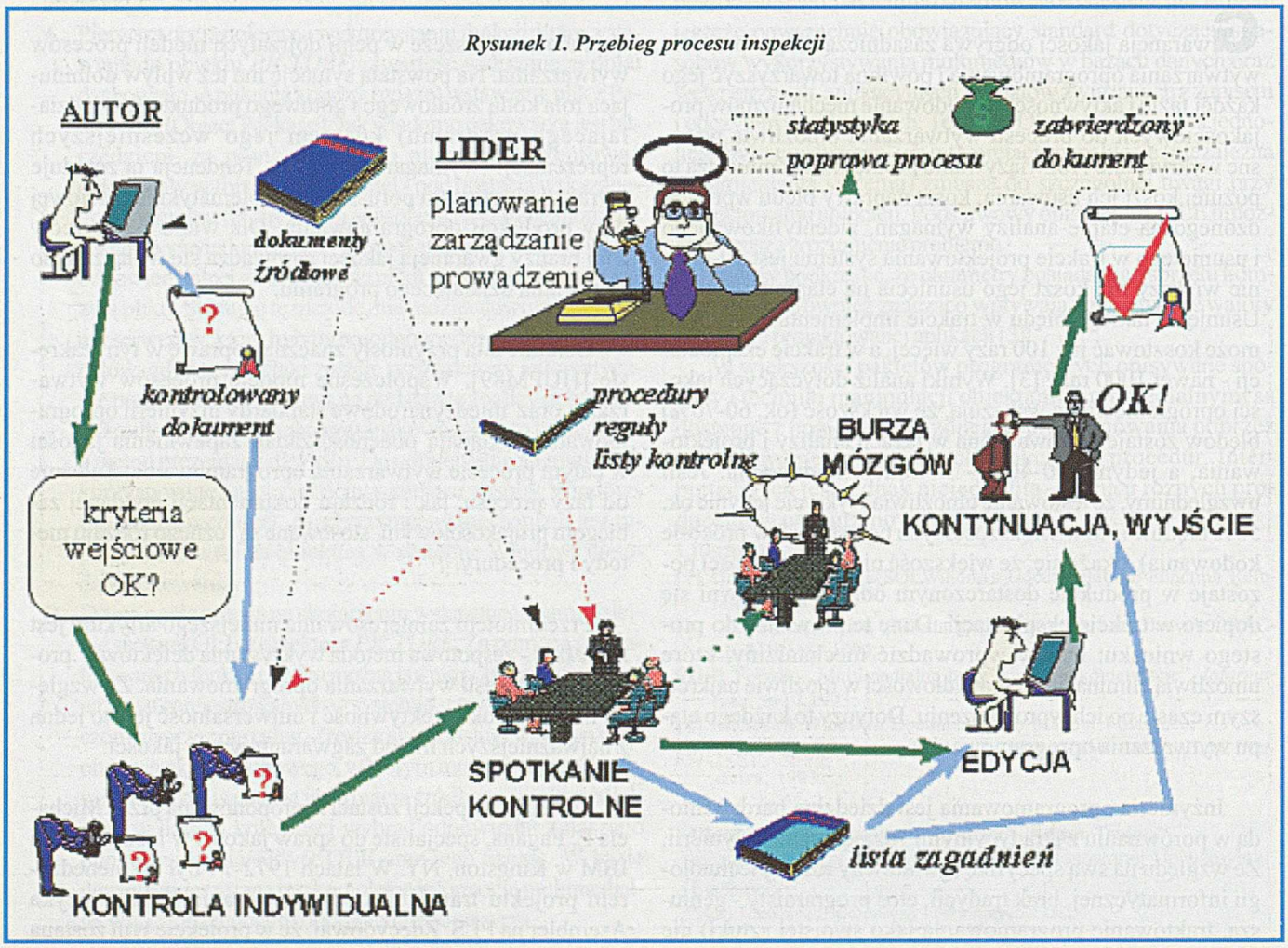
Opis techniki

Przebieg

Inspekcja, w rozumieniu Fagana, przebiega w kilku etapach (rysunek 1). Każdy z nich ma zdefiniowane cele, warunki wejściowe, wyjściowe, wykorzystywane dokumenty, produkty, przebieg oraz uczestników.

Proces inspekcji zostaje zainicjowany w chwili zgłoszenia przez autora lub dysponenta produktu zapotrzebowania na przeprowadzenie inspekcji (*faza inicjacji*). Zgłoszenie trafia do odpowiedniej komórki, która wyznacza kompetentnego lidera inspekcji. Lider sprawdza, czy produkt oraz produkty źródłowe (czyli te, na podstawie których powstał

Rysunek 1. Przebieg procesu inspekcji



produkt mający zostać poddany inspekcji) spełniają określone *kryteria wejścia*. Celem tej fazy (*faza wejścia*) jest niedopuszczenie do inspekcji produktów nieprzygotowanych, których analiza byłaby jedynie stratą czasu zespołu fachowców. *Faza planowania* obejmuje prace przygotowawcze, np. ustalenie liczby spotkań kontrolnych, wyznaczenie uczestników, przygotowanie harmonogramu prac, itp. *Spotkanie inicjujące* ma na celu wyjaśnienie wszystkim uczestnikom inspekcji celów, harmonogramu, oczekiwań w stosunku do nich i ich pracy, dystrybucję materiałów, przydział ról, wyjaśnienie procedur inspekcji, określenie parametrów wydajnościowych (np. tempa kontroli indywidualnej). W przypadku doświadczonych zespołów można z takiego spotkania zrezygnować. *Kontrola indywidualna* produktu wykonywana jest osobno przez każdego z kontrolerów, którzy opierają się na produktach źródłowych (na podstawie których został utworzony badany produkt), procedurach kontroli, regułach wytworzenia produktu i listach kontrolnych. Głównym celem pracy inspektora jest znalezienie jak największej liczby potencjalnych defektów (zagadnień) w produkcie, głównie przez analizę rozbieżności pomiędzy nim a jego dokumentami źródłowymi. Zagadnienia powinny być identyfikowane na podstawie udostępnionych dokumentów, procedur, list kontrolnych, wiedzy i doświadczenia oraz dokumentowane w sposób umożliwiający łatwe odwoływanie się do nich w trakcie spotkania kontrolnego. Optymalne tempo kontroli indywidualnej (liczba jednostek, np. stron dokumentacji, poddanych kontroli na jednostkę czasu) jest narzucone przez lidera i stanowi jeden z parametrów całego procesu umożliwiający dostosowywanie go do konkretnych sytuacji. *Spotkanie kontrolne* stanowi kluczową fazę procesu inspekcji i jest często błędnie z nią utożsamiane. W rzeczywistości jest to etap „zbierania plonów" indywidualnej pracy kontrolerów, co stanowi podstawowy cel spotkania: *udokumentowanie zagadnień zidentyfikowanych podczas sesji kontroli indywidualnych*. Zagadnienia mogą być klasyfikowane w różnych kategoriach (tabela 1).

Drugim celem spotkania kontrolnego jest *znajdowanie i dokumentowanie nowych zagadnień*, pojawiających się w trakcie dyskusji nad badanym produktem. Trzeci cel to *zgłaszanie i dokumentowanie sugestii poprawy* samego procesu inspekcji, list kontrolnych, procedur i reguł. Adresatem tych zagadnień jest komórka odpowiedzialna za inspekcje. Spotkanie kontrolne, prowadzone przez moderatora (najczęściej jest nim lider inspekcji) powinno trwać nie więcej niż dwie godziny. W przypadku obszernych produktów przewiduje się i planuje większą liczbę spotkań w ramach jednej inspekcji.

Kolejnym krokiem inspekcji jest *edycja*, która ma na celu rozwiązanie problemów udokumentowanych w trakcie spotkań kontrolnych. Dokonywana jest przez osobę pełniącą rolę edytora, którą jest najczęściej sam autor produktu. Edytor jest odpowiedzialny za uwzględnienie wszystkich spisanych zagadnień udokumentowanych podczas spotkań kontrolnych. Może to polegać na:

- modyfikacji produktu;
- wzbogaceniu dokumentu o odpowiednie komentarze umożliwiające jego jednoznaczną interpretację;
- udokumentowanym pominięciu zagadnień nie wymagających, zdaniem autora, poprawy.

Tabela 1. Przykładowa klasyfikacja zagadnień

KATEGORIA	SYMBOL	OBJAŚNIENIE
WAŻNOŚĆ	G	zagadnienie główne, o znaczącej roli dla produktu, prowadzące do jego dyskwalifikacji
	P	zagadnienie podrzędne, o niewielkim wpływie na jakość i/lub funkcjonalność produktu
	D	zagadnienie drobne, o pomijalnym znaczeniu
TYP	R	brak jakiegoś elementu, fragmentu, definicji itp.
	Ł	błędny element, fragment, rozdział, definicja itp.
	N	element nadmiarowy, np. powtórzona definicja
	Z	niejednoznaczność, sformułowanie posiadające wiele interpretacji
ROLA	!	defekt
	?	pytanie do autora, dotyczące najczęściej intencji projektowej
	+	sugestia poprawy produktu lub procesu
ZAKRES	ort	błąd ortograficzny
	log	błąd logiczny
	int	interfejs
	wyd	wydajność
	fun	funkcjonalność
	std	standard

Edytor może również zredagować sugestie poprawy produktów źródłowych.

Etap *kontynuacji* obejmuje wiele działań podejmowanych głównie przez lidera inspekcji: sprawdzenie wyników fazy edycji, w szczególności korekcji zagadnień zakwalifikowanych jako defekty główne, rozesłanie odpowiednim osobom sugestii zmian produktów źródłowych oraz propozycji poprawy procesu inspekcji.

Kolejnym krokiem jest *wyjście*, będące formalnym zatwierdzeniem przez lidera wyników inspekcji w odniesieniu do kryteriów wyjścia. Jeśli dokument był podzielony na fragmenty, każdy z nich musi podlegać procedurom oceny spełnieniu tych kryteriów. Po zakończeniu tego kroku produkt zostaje oficjalnie udostępniony do dalszego wykorzystania, z podaniem szacunkowej liczby pozostałych w nim defektów.

Dodatkowym etapem inspekcji może być tzw. *burza mózgów*, zwoływana w celu podjęcia próby identyfikacji przyczyn powstania głównych zagadnień. Zwykle prowadzi to do sugestii poprawy procesu wytwarzania oprogramowania. Obejmuje osoby uczestniczące w spotkaniu kontrolnym i trwa nie więcej niż 30 minut.

Role uczestników

W inspekcji uczestniczy zespół osób pełniących różne role. Możliwe jest spełnianie więcej niż jednej roli przez jedną osobę (np. moderator i kontroler). Osoby pełniące poszczególne role w inspekcji w różnym zakresie uczestniczą w jej fazach i aktywnościach (tabela 2). Wynika to z zespołowego i planowego charakteru procesu oraz potrzeby uzyskania wysokiej efektywności inspekcji. Instytucjonalny przewodnik inspekcji powinien zawierać rozdziały poświęcone każdej z tych ról, szczegółowym zadaniom w poszczególnych fazach oraz zakresom odpowiedzialności. Poniżej krótko omówiono poszczególne role.

Lider

Lider jest centralną postacią inspekcji. Od niego w dużej mierze zależy jej jakość i efektywność. Ponieważ inspekcja angażuje kilka osób w dłuższym czasie, jej pracochłonność zależy od sposobu jej przeprowadzenia, organizacji, przygotowania uczestników, jakości materiału podlegającego kontroli, zadanych parametrów kontroli, itp. Z tych powodów ważne jest właściwe przygotowanie osób przewidzianych do pełnienia roli lidera. Zakres odpowiedzialności lidera jest bardzo duży(tabela 1). Obejmuje kontrolę spełnienia warunków wejścia dokumentu, organizację spotkania inicjującego, wybór kontrolerów (w porozumieniu z autorem), przygotowanie i prowadzenie spotkania kontrolnego, klasyfikację i dokumentację zagadnień, przekazanie raportu z inspekcji działowi zapewnienia jakości dla celów statystycznych, kontrolę spełnienia warunków wyjścia dokumentu. Lider jest jednocześnie instruktorem, administratorem, menedżerem, przyjacielem, doradcą, itp. Posiada pewien zakres władzy nad innymi, co prowadzi do uruchamiania pewnych procesów i aktywności oraz zapobiegania innym niepożądanym sytuacjom.

Autor

Poza oczywistą rolą, jaką jest wytworzenie produktu podlegającego inspekcji, przewiduje się również aktywny udział autora w większości faz. Od autora oczekuje się aktywnej postawy projakościowej już na etapie wytworzenia produktu. Autor ma możliwość kontroli jego kompletności, zgodności z produktami źródłowymi oraz regułami oraz decyduje o momencie przedłożenia produktu do inspekcji. Zostaje automatycznie włączony do zespołu inspekcyjnego. Bierze udział w spotkaniu inicjującym, najczęściej prowadzi kontrolę indywidualną swojego produktu (może w skrajnym wypadku nawet w porę wstrzymać proces inspekcji po znalezieniu znacznych defektów w produkcie, dzięki czemu oszczędza czas innych), uczestniczy w spotkaniach kontrolnych. Ważne jest właściwe nastawienie autora do zgłaszanych uwag - powinien je akceptować i traktować z wdzięcznością i aprobatą. Podobnie ważne jest nastawienie pozostałych uczestników, którzy powinni rozumieć inspekcję jako technikę znajdowania defektów w produkcie a nie u jego autora! Autor pełni też kluczową rolę podczas fazy edycji produktu - właściwe potraktowanie każdego zapisanego zagadnienia jest jednym z warunków powodzenia całego przedsięwzięcia.

Sekretarz

Zadaniem sekretarza jest dokumentowanie zgłaszanych zagadnień w trakcie spotkania kontrolnego. Rolę tę może pełnić każdy z jego uczestników. Zaleca się jednak zaangażowanie specjalnej osoby, spoza grupy kontrolerów, co umożliwia im pełne skupienie się na zgłaszanych zagadnieniach oraz utrzymywanie odpowiedniego tempa spotkania. Wskazane jest, aby sekretarz był osobą rozumiejącą zapisywane zagadnienia - ma on wtedy możliwość scalania na bieżąco zgłaszanych zagadnień, wykrywania nieścisłości sformułowań, itp.

Kontroler

Główne fazy aktywności kontrolera to: kontrola indywidualna oraz spotkanie kontrolne. Zadaniem kontrolera jest znalezienie jak największej liczby defektów, których nie znajdą inni kontrolerzy oraz ich czytelne udokumentowanie. Kierując się sugestiami autora, należy wybierać kontrolerów spośród: autorów produktów źródłowych oraz innych osób, które mogłyby być autorami tego produktu, przyszłych odbiorców i użytkowników produktu, ekspertów i osób nim zainteresowanych.

Dokumenty pomocnicze i formularze

Dla prawidłowego i efektywnego przeprowadzenia inspekcji niezbędne są różnego rodzaju dokumenty pomocnicze

FAZA/AKTYWNOŚĆ	LIDER	AUTOR	SEKRETARZ	KONTROLER
Wytworzenie produktu		!		
Inicjacja i planowanie	!	!		
Wejście	!			
Spotkanie inicjujące	!	*	*	+
Kontrola indywidualna	!	*		!
Spotkanie kontrolne	!	+	+	+
Burza mózgów	!	+	+	+
Edycja		!		
Kontynuacja	!			
Wyjście	!	+		*
Metryki i statystyka	!			+

Oznaczenia:
! - odpowiedzialność za fazę/aktywność
\+ - udział
* - możliwość uczestnictwa w fazie/aktywności

Tabela 2. Role uczestników inspekcji

oraz formularze. Ze względu na ograniczoną objętość niniejszego artykułu zostaną one przedstawione w formie skrótowej, z podziałem na trzy kategorie.

A. Dokumenty wspomagające proces inspekcji

1. Przewodnik inspekcji - zawiera definicję wzorcowego dla danej instytucji procesu inspekcji, w ujęciu ogólnym oraz z punktu widzenia jej uczestników. Jest częścią materiałów szkoleniowych oraz wspomaga uczestników w procesie inspekcji.

2. Jednostronicowy wyciąg przewodnika inspekcji, umożliwiający szybką orientację w procesie.

3. Procedura kontroli - dokument sterujący kontrolą, zawierający opis czynności prowadzących do usunięcia błędów.

B. Dokumenty pomocnicze związane z konkretną inspekcją

1. Reguła - dokument zawierający zasady tworzenia produktu, który podlega inspekcji. W trakcie kontroli służy on jako kryterium oceny produktu w odniesieniu do produktów źródłowych.

2. Lista kontrolna - interpretacja reguł podana w formie listy zagadnień, ułatwiającej odnajdowanie nieprawidłowości.

C. Formularze

Stanowią element instytucjonalnego standardu inspekcji. Są pomocne w trakcie organizacji inspekcji, ułatwiają zbieranie danych o jej przebiegu (w tym metryk) oraz dokumentowanie. Poniżej omówiono zawartość kilku podstawowych formularzy.

1. *Zgłoszenie potrzeby inspekcji* - jest to formularz inicjujący inspekcję, częściowo wypełniany przez autora, a uzupełniany przez lidera. Zawiera takie elementy, jak: nazwa produktu, produktów źródłowych, ich autorów, pożądany termin zakończenia inspekcji, datę, warunki wejścia, decyzję lidera o dopuszczeniu produktu, itp.

2. *Plan inspekcji* - jest to formularz wypełniany przez lidera. Zawiera, oprócz informacji identyfikacyjnych, dużo danych organizacyjnych, takich jak: lista uczestników i ich role, terminy poszczególnych faz, wykaz obowiązujących reguł i list kontrolnych, zakładane tempo kontroli indywidualnej oraz grupowej.

3. *Lista zagadnień* stanowi dokumentację wszystkich zagadnień zgłoszonych podczas spotkania kontrolnego, obejmującą ich precyzyjną identyfikację, opis, klasyfikację oraz autorstwo. Przy każdym zagadnieniu zostaje pozostawione miejsce na komentarz osoby prowadzącej edycję produktu.

4. *Podsumowanie inspekcji* - jest to formularz zawierający wiele pól przeznaczonych do wypełnienia przez lidera, obejmujących: wyniki kontroli indywidualnej, spotkań kontrolnych, aktywności kontynuacji i wyjścia. Formularz ten zawiera, w formie zagregowanej, wyniki przeprowadzonych pomiarów efektywności, produktywności, liczby defektów, itp.

Metryki związane z inspekcją

Jak podano we Wprowadzeniu, poprawa produktu nie jest jedynym celem inspekcji. Osiągnięcie celów związanych z *poprawą procesu wytwarzania oprogramowania* (w tym samego procesu inspekcji) wymaga podjęcia dodatkowego wysiłku zbierania, dokumentowania i analizowania dużej ilości informacji niezwiązanych bezpośrednio z badanym produktem. Są to informacje o charakterze jakościowym, np. przyczyny pojawienia się różnego rodzaju nieprawidłowości w produkcie oraz ilościowym, np. liczba znalezio-

nych głównych zagadnień na jednostkę czasu kontroli indywidualnej, tempo dokumentowania zagadnień podczas spotkania kontrolnego, pracochłonność całego procesu. Zbieranie tych danych powinno być „wpisane" w sam proces i powinno mu towarzyszyć we wszystkich jego fazach. Takie podejście powinno być nie tylko stałym "dodatkiem" towarzyszącym samej inspekcji - powinno być trwałym elementem kultury informatycznej. Metryki stanowią bowiem bogate źródło informacji o każdym procesie, jego słabych i silnych punktach oraz stanowią bazę do jego poprawy czy optymalizacji [2] [7].

W przypadku inspekcji interesujące są grupy metryk związanych z *pracochłonnością* (całościową w rozbiciu na fazy, osoby, itp.), *wydajnością* (np. pracochłonność wykrycia w przeliczeniu na defekt, jednostkę dokumentu, osobę), *produktywnością* (np. liczba wykrytych defektów na jednostkę informacji, w rozbiciu na kontrolę indywidualną, spotkania kontrolne) czy *skutecznością* (szacowany procent wykrytych defektów). Interesujący jest wynik analizy tych danych, zgromadzonych na podstawie większej liczby inspekcji, przeprowadzonych z różnymi parametrami początkowymi. Przykładowo, większe tempo inspekcji prowadzi do zmniejszenia kosztów (poprawa wydajności), ale jednocześnie powoduje pozostawienie w produkcie większej liczby defektów (zmniejszenie skuteczności). Posiadanie obszernej bazy danych, zawierającej wyniki pomiarów z wielu przeprowadzonych inspekcji, umożliwia skuteczniejsze zarządzanie zasobami, lepszą kontrolę i poprawę samego procesu inspekcji oraz szacowanie jakości produktu [8].

Koszty i zyski z inspekcji

Pobieżna analiza przebiegu oraz wyników pomiarów rzeczywistych inspekcji możne prowadzić do wniosków niekorzystnych dla oceny opłacalności tej techniki. Nie podlega dyskusji fakt znacznego zaangażowania grupy fachowców, dużego nakładu prac organizacyjnych, biurowych, itp. Całkowita pracochłonność przeciętnej inspekcji to dziesiątki, a nawet setki cennych roboczogodzin. Aby ją w pełni ocenić, należałoby uwzględnić koszty szkoleń, zarządzania inspekcjami, analizy metryk, eksperymentów. Jednakże dopiero całościowa ocena wyników inspekcji, uwzględniająca jej wpływ na cały proces wytworzenia produktu informatycznego, umożliwia wyciągnięcie właściwych wniosków.

Stosując statystyczne metody opracowywania wyników pomiarów zbieranych w trakcie inspekcji(uwzględniające również koszt naprawy błędów, które bez zastosowania inspekcji nie zostaną zidentyfikowane i ujawnią się w późniejszych etapach procesu wytworzenia lub eksploatacji), można w sposób pełny zbilansować koszty i zyski. Literatura na ten temat jednoznacznie ocenia inspekcje jako przedsięwzięcie bardzo opłacalne [8]. Poza zyskiem czysto ekonomicznym, związanym z redukcją całkowitej pracochłonności, wskazywane są inne korzyści ze stosowania tej techniki. Skoncentrujmy się na kilku wybranych zagadnieniach.

- Dzięki stosowaniu inspekcji we wszystkich fazach wytwarzania, większość defektów jest usuwanych systematycznie, w miarę ich wprowadzania do kolejnych reprezentacji produktu, a nie jedynie w trakcie jego ostatecznego testowania. Ma to duży wpływ na jakość produktu końcowego – usuwanie defektów „odziedziczonych" po poprzednich etapach może głęboko naruszyć jego ostateczną, wypracowaną już strukturę. Ogranicza się też w dużym stopniu pracochłonność szczególnie kosztownego testowania i usuwania defektów w „ostatniej chwili" przed dostarczeniem gotowego produktu;

- Systematyczne stosowanie inspekcji oraz dokumentowanie ich wyników może wspomóc program doskonalenia procesu wytwarzania przez ujawnianie jego krytycznych obszarów oraz wprowadzanie bardziej efektywnych standardów. Takie podejście prowadzi do zmniejszenia liczby błędów w dostarczonych produktach, co z kolei podnosi wiarygodność wytwórcy na rynku.

Inne korzyści wynikają z podnoszenia jakości i kwalifikacji wytwórców oprogramowania przez udział w szkoleniach, spotkaniach kontrolnych, „burzach mózgów", co prowadzi do osiągnięcia lepszej znajomości produktów wytwarzanych w ramach organizacji. Poprawiają się (pod warunkiem stworzenia właściwej atmosfery wokół problematyki jakościowej) relacje między zespołami, wydziałami i w ramach nich tworzą się warunki do lepszego wzajemnego poznania.

Poniżej podano kilka konkretnych korzyści z zastosowania inspekcji w odniesieniu do całego procesu wytwarzania oprogramowania [5]:

- wzrost produktywności o 30% - 100%;
- skrócenie czasu wytwarzania o 10% - 30%;
- 5-10-krotna redukcja czasu i kosztu testowania;
- redukcja kosztów pielęgnacji o rząd wielkości;
- nieomal "automatyczna" poprawa jakości procesu i produktu;
- eliminacja poślizgów czasowych w dostarczaniu produktów.

Warunki sukcesu inspekcji

Inspekcja nie może być traktowana w oderwaniu od systemu zapewnienia jakości instytucji wytwarzającej oprogramowanie. Muszą istnieć jasne i jawnie zdefiniowane cele jej zastosowania w całym procesie(w tym momenty czasu w ramach poszczególnych faz) produkty podlegające inspekcji, formalny proces ich dopuszczania do dalszego wykorzystania itp. Inspekcja jest więc narzędziem zapewnienia jakości, stosowanym w ramach większego programu zapewnienia jakości.

Niezależnie od zasad jej włączania do tego programu, rodzaju badanego produktu, liczby zaangażowanych osób,

I N T E R N E T KONCESJA 29/96/J

File Edit Window Help

BACK NEXT STOP HOME LOOK BEST CODE PRINT HELP

Site: http://heros.juni.com/

http://heros.juni.gliwice.pl./

1987 – 1997

JUNISOFTEX OFERUJE:

- Nowoczesne obiektowe aplikacje
- Zarządzanie procesem projektowym wsparte narzędziami CASE (w tym Select OMT)
- Analizy dziedzinowe oparte o metodę łączącą nowoczesność z 20 letnim doświadczeniem wdrożeniowym
- Sprawne i szybkie wdrożenia zgodne z obowiązującymi przepisami
- Projektowanie i wykonawstwo sieci komputerowych
- Typowe, publiczne usługi internetowe dla osób prawnych i fizycznych
- Technologie internetu do restrukturyzacji procesów organizacyjnych
- Technologie internetu w systemach wspomagających proces zarządzania
- Integracje różnych rodzajów sieci
- dostawy sprzętu komputerowego renomowanych firm
- dostawy sprzętu peryferyjnego

Perfect-Ekspert '97

PRODUKT POLSKI

System Perfect-Ekspert'97 jest zbiorem aplikacji integrujących procesy gospodarcze w całym przedsiębiorstwie.

Strategia zarządzania i systemy informowania kierownictwa
Sprawozdawczość Obserwacja Dociekanie Wnioskowanie Komentowanie Symulacje i Prognozy

System finansowo-kosztowy
Banki Kasa Sprzedaż Zakup Koszty proste Rozliczanie kosztów Sprawozdawczość

System ewidencji majątku
Ewidencja Inwentaryzacja Koszty Plan amortyzacji Przeszacowania Sprawozdawczość

Systemy obsługi klienta, obrót towarowy, sprzedaż
Zamówienia Ewidencja Inwentaryzacja Zakup towarów Kompletacja Ewidencja opakowań zwrotnych Sprzedaż Analizy Sprawozdawczość

Systemy gospodarki materiałowej i zaopatrzenia
Zakupy Ewidencja Inwentaryzacja Dyspozycja Rezerwacja Koszty Analiza Sprawozdawczość

System kadrowo-płacowy oraz dyscypliny
Dokumentacja i ewidencja Sytemy wynagrodzeń Czas pracy i rejestratory czasu pracy Kwalifikacje i szkolenia Emisja dokumentacji ZUS i zasiłki Sprawozdawczość

System przygotowania produkcji
Dokumentacja technologiczna Narzędzia i urządzenia pomiarowe Kalkulacje i ofertowanie Emisja dokumentacji warsztatowej Plan zakupów Plany warsztatowe Produkcja w toku Rozliczenie produkcji

Systemy pomiarowe i stanowiska pomiarowe
Statystyczne oceny jakości Dokumentacja jakości Świadectwa jakości i atesty Baza pomiarowa

Przedsiębiorstwo Produkcyjno–Usługowe Innowacji Technicznych

JUNISOFTEX® Sp. z o.o.

ul. Kościuszki 1c, 44-100 GLIWICE
tel.: (0-32) 230 69 31 do 33
fax: (0-32) 230 69 34
E-mail: DYREKTOR@JUNIGLIWICE.PL
E-mail: PREZES@JUNIGLIWICE.PL
Nr Polpak: 13211054
NIP: 631-020-03-11
Regon: P-008012971-94927177-59-3-671-27177
Konto: PKO I/O Gliwice 27661-225298-136

Dyrektor Naczelny – Krzysztof Niewiadomski
Prezes – Bronisław Jureczko
Kapitał zakładowy 1.150.000.000 PLZ (115.000 PLN)

Firma JUNISOFTEX obecna jest na rynku od 10 lat. W tym czasie wdrożono kilkaset dużych aplikacji. Naszej firmie zaufało jak do tej pory około 100 przedsiębiorstw produkcyjnych w tym 6 spółek giełdowych, 10 przedsiębiorstw NFI, kilka przedsiębiorstw przemysłu hutniczego i wiele innych.

Lista aktualnych użytkowników „Perfect-Ekspert"

- Elektrownia RYBNIK w Wielopolu
  jedyna w Polsce elektrownia rozliczająca koszty rzeczywiste remontów na 3 tys. obiektów
- ELEKTROBUDOWA S.A. w Katowicach
- POLIFARB S.A. w Cieszynie
- WEDEL w Warszawie
- P.P.O.W. AGROS-PIŃCZÓW w Pińczowie
- Zakłady Odzieżowe BYTOM S.A. w Bytomiu
- Maszyny Elektryczne CELMA w Cieszynie
- Fabryka Amortyzatorów KROSNO S.A. w Krośnie
- Instytut Spawalnictwa w Gliwicach
- GEOFIZYKA w Toruniu
- POLSPORT w Bielsku-Białej
- POLAM-KOSTUCHNA w Katowicach
- Walcownia Metali DZIEDZICE S.A. w Czechowicach-Dziedzicach
- ELEKTRONARZĘDZIA CELMA S.A. w Cieszynie
- Zakłady Tworzyw Sztucznych KRYWAŁD ERG w Knurowie
- Fabryka Aparatów Elektrycznych APENA S.A. w Bielsku-Białej
- MOSTOSTAL Słupca
- Fabryka Osi Napędowych w Radomsku
- Huta Szkła Okiennego KUNICE S.A. w Żarach
- Huta ŁAZISKA w Łaziskach Górnych
- ELEKTROMONTAŻ 3 S.A. w Katowicach
- Fabryka Palenisk Mechanicznych w Mikołowie
- DRAGMOR sp. z o.o. w Szczecinie
- Dąbrowskie Zakłady Naprawcze Przemysłu Węglowego w Dąbrowie Górniczej
- Centrum Naukowo Techniczne Kolejnictwa w Warszawie
- Zakłady Rowerowe ROMET Kowalewo
- SUWARY S.A. w Pabianicach
- Zakłady Naprawcze Przemysłu Węglowego w Rybniku
- Rejonowe Przedsiębiorstwo Wodociągów i Kanalizacji w Gliwicach
- Zakłady Tłuszczowe BIELMAR S.A. w Bielsku-Białej
- GZUT w Gliwicach
- MEFTA w Mikołowie
- KRONOPOL w Żarach
- Huta BANKOWA
- Walcownia Metali ŁABĘDY S.A. w Gliwicach
- Huta 1-go MAJA w Gliwicach
- ELEKTROMETAL Zakład Telemechaniki Górniczej w Cieszynie
- Zakłady Metalurgiczne TRZEBINIA w Trzebini
- Huta SILESIA w Rybniku
- Fabryka Automatyki Chłodniczej S.A. w Cieszynie
- Zakłady Górniczo-Hutnicze BOLESŁAW w Bukownie
- MIFAMA w Mikołowie

inspekcja jest (wbrew pozorom) bardzo złożonym przedsięwzięciem organizacyjnym. Jak w przypadku każdej grupowej aktywności jej sukces zależy od wielu czynników. Jednym z najważniejszych jest ścisłe zdefiniowanie całego procesu, jego faz, ról uczestników, zadań, zasad organizacji spotkań, itd. Organizacja stosująca tę technikę powinna dopracować się instytucjonalnego przewodnika, jednoznacznie definiującego te elementy.

Osobnym zagadnieniem jest proces wprowadzania inspekcji do organizacji. Musi się to odbywać według wypracowanego wcześniej scenariusza uwzględniającego takie czynniki, jak: pełne poparcie wyższych szczebli zarządzania, świadomość kadry inżynieryjno-technicznej, szkolenia, tworzenie bazy metryk, dokumentowanie i wiele innych.

Ważne są też czynniki socjologiczne, psychologiczne, relacje interpersonalne, ogólna atmosfera w organizacji. Należy zwrócić uwagę na fakt potencjalnego zagrożenia pozycji autora produktu. Nikt nie lubi być oceniany - jest to wspólna cecha większości z nas. Osiągnięcie właściwego nastawienia pracowników do szeroko rozumianej polityki projakościowej ma decydujące znaczenie dla jej sukcesu: oceniamy produkt a nie jego autora. Próby wykorzystania wyników inspekcji (w tym metryk wydajnościowych) do bezpośredniej oceny pracownika mogą wytworzyć barierę trudną do pokonania - zbiorową niechęć czy nawet „obronę" przed jakąkolwiek formą kontroli i zapewnienia jakości.

Inne ważne czynniki wpływające na sukces inspekcji, to:

- efektywne prowadzenie spotkań kontrolnych, w tym niedopuszczanie do dyskusji w trakcie spotkań, szczególnie na temat znalezionych rozwiązań (typowa skłonność fachowców!); nie można dopuścić do tego, aby dyskusja (nawet najmądrzejsza) między autorem a jednym z kontrolerów doprowadziła do „konsumpcji" cennego czasu kilku innych osób;
- atmosfera spotkań - musi sprawiać przyjemność uczestnikom i nie prowadzić do napięć i niechęci;
- forma oraz dostępność dokumentacji podlegającej inspekcji, źródeł, standardów, reguł, procedur, list kontrolnych;
- jasne kryteria wejścia i wyjścia;
- przeszkolenie uczestników, dostępność przewodników dotyczących poszczególnych ról i etapów inspekcji.

Do zagrożeń inspekcji, oprócz niespełnienia powyższych założeń, należeć mogą:

- prowadzenie inspekcji w sposób niefachowy i nieefektywny;
- niewystarczające kwalifikacje kontrolerów;
- niska jakość oraz ograniczony zakres kontroli indywidualnej;
- skłonność autora do lekceważenia defektów na etapie opracowywania dokumentów („inspekcja i tak wskaże błędy..."");
- nieuzasadniona obrona autora własnych rozwiązań spowodowana poczuciem zagrożenia;
- krytyczne nastawienie uczestników inspekcji do autora;
- spadek motywacji wraz z upływem czasu spotkania kontrolnego (wyzwanie dla lidera!).

Inspekcja jest bardzo efektywnym narzędziem podnoszenia jakości procesu wytwarzania oprogramowania. Stosowana we większości faz życia systemu informatycznego pozwala na wczesne wykrywanie i usuwanie defektów w produktach wytwarzanych w poszczególnych fazach. Ograniczenie zjawiska propagacji defektów przez kolejne reprezentacje systemu redukuje konieczność bardzo kosztownego ich usuwania w późniejszych fazach. Osiągnięcie takich wyników zależy jednak od właściwego wprowadzenia tej techniki w organizacji, jak również jej poprawy w ramach nieustannego doskonalenia całego procesu wytwarzania. Sposób prowadzenia inspekcji, stosowane procedury, listy kontrolne, formularze, klasyfikacja zagadnień, normy wydajnościowe zależą od specyfiki organizacji i muszą być elementem jej polityki projakościowej.

Literatura

[1] ANSI/IEEE Standard 729-1983 - IEEE Standard Glossary of Software Engineering Terminology;

[2] J. Barnard, A. Price, ***Managing Code Inspection Information***, IEEE Software, March 1994, pp. 59-69;

[3] B. W. Boehm, ***Software Engineering Economics***, Prentice-Hall, 1981;

[4] M. E. Fagan, ***Design and Code Inspections to Reduce Errors in Program Development***, IBM systems Journal, 15(3), pp. 182-211;

[5] T. Gilb, D. Graham, ***Software Inspection***, Addison-Wesley Publishing Company, 1993;

[6] W. S. Humphrey, ***Managing the Software Process***, Addison-Wesley, 1989;

[7] F. J. Redmill, E. A. Johnson, B. Runge, ***Document Quality-Inspection***, British Telecommunications Engineering, Vol. 6, Jan, 1988, pp. 250 - 256;

[8] E. F. Weller, ***Lessons from Three Years of Inspection Data***, IEEE Software, Sept. 1993, pp. 38-45.

Marek Jurkowlaniec jest pracownikiem PZF POLFA S.A. w Poznaniu.

BjØrn Runge Dansk Data Elektronik A/S, Dania.

weź pojedź na wycieczkę

udział w konkursie
na najlepsze **rozszerzenie**
dla Microsoft Office 97 i...

do ciepłych krajów.

Zapraszamy Cię do wzięcia udziału w konkursie na najlepsze rozszerzenie dla Microsoft Office 97. Rozszerzeniami mogą być makropolecenia, szablony, programy – cokolwiek sobie wymarzysz, z jednym zastrzeżeniem: muszą być opracowane na bazie Microsoft Office 97. W konkursie może wziąć udział każdy, kto napisze rozszerzenie i spełni pozostałe warunki konkursu.*
Na autora najlepszego rozwiązania czeka egzotyczna wycieczka dla dwóch osób.
Oprócz tego do wygrania m.in. telefony komórkowe, komputery, urządzenia zewnętrzne do komputerów, oprogramowanie i wiele innych nagród ufundowanych przez partnerów firmy Microsoft.**
Najlepsze rozszerzenia będą promowane przez Microsoft i magazyn CHIP.
Konkurs trwa od 1 października 97 do 30 listopada 97. Ogłoszenie wyników nastąpi **22 grudnia 97** w Gazecie Wyborczej.
Więcej informacji na temat konkursu można znaleźć w magazynie CHIP.
A zatem do dzieła!

* Pełny regulamin konkursu znajdziesz pod numerem Infoserwisu oraz na stronie internetowej.

** Pełna lista nagród znajduje się w regulaminie konkursu.

Zapraszamy do odwiedzenia polskiej strony WWW firmy Microsoft:
www.microsoft.com/poland/office97/konkurs/

CHIP
Magazyn komputerowy

Microsoft®
WHERE DO you WANT TO GO TODAY?

© 1997 Microsoft Corporation. Wszelkie prawa zastrzeżone. Microsoft jest zastrzeżonym znakiem towarowym.
Dodatkowe informacje - Infoserwis Microsoft tel. (0 22) 865 99 33.
Bezpłatna pomoc techniczna dla zarejestrowanych użytkowników Microsoft Office tel. (0 22) 865 99 66.
Linia antypiracka BSA tel. (0 22) 661 55 12.

ALMA integrator systemów informatycznych
era GSM
HEWLETT® PACKARD
OPTIMUS SA
VADIM
ZETO ZETO - KIELCE S.A.

POLSKIE TOWARZYSTWO INFORMATYCZNE

BIULETYN

NUMER 8 (137) ROK XVI LISTOPAD 1997

Gdy zbliża się jesień, w PTI mówi się przede wszystkim o Jesiennych Spotkaniach w Mrągowie. W tym roku jest podobnie. Dlatego aktualny Biuletyn w znacznej części poświęcony jest temu wydarzeniu. Zapraszamy do Mrągowa zarówno stałych bywalców, jak i nowe twarze.

Redakcja

Trzynaste Jesienne Spotkania PTI

Mrągowo '97, 24-28 listopada 1997

Informatyka w organizacjach — stan dzisiejszy i rzut oka w przyszłość

Kolejne spotkanie Polskiego Towarzystwa Informatycznego jest kontynuacją Szkół Jesiennych PTI, które odbywały się w Mrągowie począwszy od roku 1984 oraz Forów Technologii Informatycznych, odbywających się w tym samym miejscu w latach 1991-1995. W swej trzynastoletniej historii konferencje mrągowskie były okazją do kontaktu przedstawicieli różnych środowisk związanych z informatyką: naukowców, pracowników przedsiębiorstw informatycznych, przedstawicieli władzy.

Każdy dzień konferencji będzie podzielony na cztery pasma: Prezentacje (1 i 2), Wykłady i Panele. Pasmo **Wykłady** służy przedstawieniu stanu badań podstawowych prowadzonych w polskich uczelniach oraz omówieniu "nowinek" nauki światowej. Wykłady będą budować główną linię merytoryczną konferencji. Do treści wykładów nawiążą **Prezentacje**. W trakcie tego cyklu wystąpią producenci środków informatyki: sprzętu i oprogramowania. W ten sposób przy utrzymaniu aspektu praktycznego konferencji uzyskana zostanie jedność tematyczna: prezentacje producentów będą unaocznieniem tez wykładów naukowych. Pasmo **Panele** będzie zamykało dzień pozwalając uczestnikom konferencji na dyskusję na główne tematy dnia, jak również na tematy bardziej ponadczasowe.

Opisana wyżej struktura będzie — mamy nadzieję — służyła wieloaspektowej i pogłębionej dyskusji wiodącego tematu tegorocznej konferencji. Informatyka jest stosowana powszechnie w wielu rodzajach i typach organizacji. Złożoność współczesnych rozwiązań jest tak duża, że wszyscy, z profesjonalistami na czele, czują się nieco zagubieni. Z drugiej strony dopiero spojrzenie z poziomu ogólnego pozwala znaleźć jednak niezmienniki procesów rozwoju informatyki. Zamiarem Komitetu Programowego było przygotowanie konferencji, w której przechodząc od szczegółu (środki techniczne i rozwiązania informatyki prezentowane głównie w paśmie prezentacyjnym) do ogółu (wykłady monograficzne i dyskusje panelowe) uczestnik będzie w stanie lepiej zrozumieć tendencje rozwojowe zastosowań informatycznych i techniki przetrwania w coraz szybciej zmieniającym się świecie zastosowań.

Tak skonstruowany program adresowany jest przede wszystkim do osób zarządzających zastosowaniami informatyki w organizacjach oraz do tych, którzy dostarczają organizacjom rozwiązania informatyczne. Równocześnie konferencja jest dobrą okazją do lepszego wzajemnego zrozumienia tych grup pozostających na co dzień w naturalnym konflikcie. Proporcja ogólnego do szczegółowego pozwala na skorzystanie z konferencji zarówno osobom na najwyższych stanowiskach, jak również tym, których praca nosi bardziej techniczny charakter.

Organizatorzy nie zapomnieli również o kulturowym i integracyjnym aspekcie spotkania. Hotel Mrongovia, w którym odbywa się konferencja zapewnia luksusowe warunki pobytu. Oprócz basenu i sauny, są w nim również liczne bary, dyskoteka i oddane w tym roku do użytku nowe centrum kongresowe z nowoczesnym wyposażeniem. Koncert zespołu Wolna Grupa Bukowina będzie okazją do zwiedzenia innego niż informatyczny, ale chyba nawet bardziej interesującego świata.

Posiedzenie Zarządu Głównego PTI

W dniu 15 października 1997 roku odbyło się w siedzibie InfoVide w Warszawie posiedzenie Zarządu Głównego PTI. W programie posiedzenia znalazły się między innymi:

- informacje Prezesa na temat dobrze rozwijającej się współpracy PTI ze Stowarzyszeniem Księgowych w Polsce,
- sprawy organizacji Biura ZG, referowane przez Sekretarza Generalnego, w tym zapewnienie o powołaniu nowego opiekuna strony WWW Towarzystwa, kol. Trawki z Katowic - efekty Jego pracy będą widoczne niebawem,
- omówienie stanu akcji wprowadzania ECDL w Polsce: pod względem liczby rozprowadzonych kart EKUK Polska jest na trzecim miejscu w Europie po Szwecji i Austrii; ponieważ akcja będzie się rozwijać dalej, powołano Komisję Autoryzacji Programów Szkolenia ECDL pod przewodnictwem Kol. Zygmunta Mazura i postanowiono udzielać autoryzacji na okres dwóch lat,
- przyjęcie pięciu nowych członków PTI,
- omówienie programu Trzynastych Jesiennych Spotkań PTI Mrągowo «97
- dyskusja nad usprawnieniem sprawozdawczości finansowej - ustalono, że ZG otrzymywać będzie informacje miesięczne i kwartalne od Oddziałów i Kół,
- przyjęto, że w związku z obchodami 50-lecia informatyki polskiej, przyszłoroczne edycje Szkół w Świnoujściu, Szczyrku i Mrągowie poświęcone będą tej rocznicy
- ustalono, że składka członkowska za 1998 rok wynosić będzie 40 zł (dla studentów - 20 zł)

Przypominamy, że członkowie PTI mogą prenumerować Informatykę, a tym samym również Biuletyn PTI, ze zniżką 50%.

PROGRAM

DATA	PREZENTACJE PRZEDPOŁUDNIOWE			WYKŁADY	PREZENTACJE POPOŁUDNIOWE		DYSKUSJE PANELOWE	PO KOLACJI
	8:45-9:30	9:45-10:30	11:00-11:45	12:00-13:30	15:00-15:45	16:00-16:45	17:15-18:45	20:30
Poniedziałek 24.11.97	**Baan** Beata Michalska *Dynamiczne zarządzanie przedsiębiorstwem w systemie BAAN IV*	**SAP** Jacek Mendocha *SAP R/3 — Nowe Technologie*	**Computron** Roman Burzyński *Księgowość zarządcza dla dużych organizacji — rozwiązanie firmy Computron*	**prof. Roman Słowiński** *Inteligentne systemy wspomagania decyzji*	**IBM** Michał Jędrusik, Grzegorz Kolecki *S390 na dziś i jutro*		*Czy za darmo to za drogo?* **moderator: Bogusław Jackowski**	
Wtorek 25.11.97	**CompFort Meridian** Mariusz S. Kaźmierski *Oprogramowanie BMC PATROL do zarządzania aplikacjami w środowisku otwartym; Oprogramowanie BMC dla środowiska DB2 mainframe*		**OCÉ** Mirosław Tarczoń *Wysokowydajne systemy drukujące — OCÉ Printing Systems*	**prof. Janusz Górski** *Inżynieria wymagań, czyli jak unikać zbędnej pracy?*	**Comparex** Josh Krisher *Pamięci dyskowe dziś i jutro*		*Sprzeczności w społecznym odbiorze informatyki* **moderator: Tomasz Kulisiewicz**	
Środa 26.11.97	**Computer Associates** Maciej Rowiński *Zarządzanie systemami informatycznymi*		**Lotus** Andrzej Oborzyński *Lotus Notes/Domino — system otwarty*	**Borys Stokalski** *Informatyka w czasach zarazy. Zarządzanie ryzykiem w projektach informatycznych*	**Lipro Sp. z o. o.** Dr Dieter Küchler *IloS — zdecentralizowany system zarządzania czwartej generacji dla post-taylorystycznych przedsiębiorstw*		**Novell** Jacek Pacholczyk *Strategia firmy Novell dotycząca usług sieciowych*	Koncert zespołu **Wolna Grupa Bukowina**
Czwartek 27.11.97	**Microsoft** Jacek Myrcha, Ryszard Dałkowski *Microsoft Small Business Server — serwer dla małych i średnich firm*		**MHM Computer** Katarzyna Stefaniak, Andrzej Wyrzykowski *Macierze dyskowe Symmetrix — nowe spojrzenie na pamięci masowe*	**dr Krzysztof Nałęcki** *Weblikacje, webifikacja. Komu to służy?*	**Informix** Tadeusz Ostrowski *Informix-Universal Web architecture*	**Xerox** Richard Schupp *Wartość dokumentu*	*10000_2 lat PTI* **moderator: prof. Zdzisław Szyjewski**	
Piątek 28.11.97	**CIM-mes Sp. z o. o.** Krzysztof Grabowiecki, Thierry Audet *Oprogramowanie CAS/CADE firmy* MATRA Datavision *do budowania aplikacji bazodanowych i graficznych 3D*	**ZETO OLSZTYN Sp. z o.o.** Jerzy Kurowski *SALVO — oprogramowanie firmy SIMWARE — dostęp do aplikacji i korporacyjnych baz danych z sieci INTERNET/INTRANET*		**Andrzej Odyniec** *Internet dla każdego*				

Nowi Członkowie PTI przyjęci na posiedzeniu ZG PTI w dniu 15 listopada 1997 r.

1. Tadeusz Chełkowski - student ze Szczecina
2. Marcin Jackowski - student z Polic
3. Krzysztof Grabowski - student z Sopotu
4. Grzegorz Warisela z Bogatyni
5. Beata Fularska z Kielc

Aby wziąć udział w XIII Jesiennych spotkaniach PTI - Mrągowo '97 należy przesłać na adres Biura Konferencji:

PTI Oddział Górnośląski
40-012 Katowice ul. Jana 10
tel./fax: +48-(32)-253-96-78
e-mail: ptiog@netpol.pl

opatrzone **pieczęcią instytucji wysyłającej** zgłoszenie uczestnictwa o treści podanej poniżej:

KARTA UCZESTNICTWA

Zgłaszamy udział w Trzynastych Jesiennych Spotkaniach PTI - Mrągowo '97

Imię i nazwisko .....

Tytuł i stanowisko .....

Adres do korespondencji .....

(tel./fax) ..... *e-mail* .....

Podpis uczestnika .....

Potwierdzamy, że nasza instytucja pokryje koszty uczestnictwa i materiałów konferencyjnych, w łącznej wysokości (właściwe zakreślić, ceny w nowych złotych):

	Członkowie PTI	Osoby nie będące członkami PTI
Wpłata do 10. 11. 1997	400,-	450,-
Wpłata po 10. 11. 1997	450,-	500,-

Ponadto nasza instytucja zobowiązuje się do pokrycia we własnym zakresie kosztów podróży i zakwaterowania pracownika, w wysokości 130 PLN za dobę wraz z całodniowym wyżywieniem w pokoju jednoosobowym ☐ lub 100 PLN w dwuosobowym ☐. Pobyt:

23/24.11.1997 ☐ 24/25.11.1997 ☐ 25/26.11.1997 ☐ 26/27.11.1997 ☐ 27/28.11.1997 ☐

Łączną kwotę kosztów uczestnictwa i zakwaterowania przekażemy na konto:

PTI Oddział Górnośląski PKO BP II O/Katowice
10202326-701343-270-222

Do ceny powyższej należy doliczyć 22% PTU. W przypadku rezygnacji z udziału zobowiązujemy się do pokrycia kosztów uczestnictwa.

Upoważnienie:

Upoważniamy Polskie Towarzystwo Informatyczne Oddział Górnośląski, NIP 522-000-20-38, 40-012 Katowice, ul. Jana 10, tel/fax +48 (32) 253 96 78 do wystawienia faktury VAT/ rachunku uproszczonego za uczestnictwo, bez podpisu odbierającego:*

firma .....

NIP .....

Główny Księgowy Dyrektor

*Niepotrzebne skreślić

Softarg'97

Między 30 września a 3 października odbyły się w Katowicach XI Międzynarodowe Targi Oprogramowania Softarg'97 pod merytorycznym patronatem Polskiego Towarzystwa Informatycznego. Patronatu honorowego podjął się Minister Spraw Wewnętrznych i Administracji, natomiast patronat prasowy objęła „Gazeta Wyborcza". Tegoroczny Softarg zdaje się zaprzeczać poglądom, jakoby formuła tej imprezy i określony asortyment wystawienniczy sprowadzały go jedynie do poziomu indywidualnych odbiorców i małych firm. W tym bowiem roku targi odbywały się ze szczególnym uwzględnieniem oferty dla urzędów administracji publicznej, instytucji i przedsiębiorstw, co znalazło wyraz zarówno w ekspozycji, jak i w specjalnym cyklu seminariów. Ponadto miejsce swoje znalazło tutaj wiele produktów oprogramowania o niezwykle szerokim wachlarzu zastosowań: od sterowania procesem produkcyjnym, systemów wyszukiwania informacji, systemów zarządzania przedsiębiorstwem po oprogramowanie narzędziowe i edukacyjne. Podstawową ofertę firm wzbogacały ich pokazy własne oraz liczne seminaria. W Softargu'97 brało udział ponad 240 firm krajowych i zagranicznych, między innymi z USA i Republiki Czeskiej.

W bogatej gamie wystawianych produktów trudno było dopatrzeć się takiego, który dyskwalifikowałby pozostałe. Ogólnie wysoki poziom oprogramowania utrudniał wytypowanie zdecydowanych liderów i prawdopodobnie w przyszłych latach proces ten będzie się utrwalał, pozostawiając różnice w zastosowaniach, a niwelując je na poziomie warsztatowym.

Licząc na lepszą aurę i niezgorszą ofertę wystawców w przyszłym roku, informuję że XII Międzynarodowe Targi Oprogramowania Softarg'98, odbędą się w dniach 08 - 11 września 98r.

Piotr Kowalski

Redaktor: EWA ŁUKASIK
e-mail: lukasik@poznlv.put.poznan.pl, tel. (61) 782 373,
Instytut Informatyki, Politechnika Poznańska,
ul. Piotrowo 3a, 60-965 Poznań

Polskie Towarzystwo Informatyczne, Zarząd Główny
tel. (22) 624 60 61 w.328, tel./fax (22) 652 32 59
URL: http://www.pol.pl/pti
ul. Żelazna 87, 00-879 Warszawa

movex zintegrowany system klasy MRP II

rozwój produktu
- zarządzanie danymi produktu
- zarządzanie zmianami konstrukcyjnymi
- zarządzanie dokumentacją techniczną
- zarządzanie projektami rozwoju
- zarządzanie danymi materiałów specjalnych

sprzedaż
- obsługa ofert sprzedaży
- obsługa ofert usług
- obsługa ofert umów serwisowych
- zarządzanie projektami sprzedaży
- zarządzanie potencjalnymi klientami
- umowa sprzedaży
- umowa serwisowa
- administrowanie kampaniami

realizacja zamówienia
- obsługa zamówień sprzedaży
- obsługa zleceń projektowych
- obsługa zleceń serwisowych
- obsługa harmonogramów dostaw
- konfiguracja produktu
- raportowanie wysyłek
- transport
- fakturowanie
- obsługa wezwań do zapłaty
- obsługa przyjęcia zapłaty

zaopatrzenie
- obsługa ofert
- kalkulacja kosztów zakupu
- obsługa zamówień zaopatrzeniowych
- obsługa harmonogramu dostaw
- przyjęcie towarów
- zarządzanie reklamacjami
- obsługa faktur dostawców
- płatności

produkcja i serwis
- dane o zasobach
- obsługa zleceń produkcyjnych
- harmonogramowanie
- zlecenia produkcyjne do wykonania
- kalkulacja kosztów zlecenia produkcyjnego
- zarządzanie zdolnościami produkcyjnymi
- obsługa zleceń serwisowych

dystrybucja
- dane o zakładach i magazynach
- obsługa zleceń dystrybucyjnych
- raportowanie zleceń dystrybucyjnych
- rezerwacja transportu
- zarządzanie opakowaniami
- zarządzanie wysyłkami
- planowanie zdolności transport.

obsługa posprzedażna
- obsługa zleceń serwisowych
- obsługa reklamacji
- obsługa zwrotów
- zarządzanie fakturami korygującymi
- umowy gwarancyjne

zarządzanie strategiczne
- efektywność realizacji zamówień
- badanie satysfakcji klientów
- statystyka sprzedaży i klientów
- statystyka zakupów i dostawców
- pomiar efektywności dostawców
- statystyka produkcji
- statystyka elementów
- analiza woluminu sprzedaży
- analiza marży pokrycia
- analiza zapasów

finanse
- księga główna
- budżetowanie
- zarządzanie przepływem pieniądza
- rachunek kosztów
- środki trwałe
- zobowiązania
- należności
- kalkulacja kosztów produktu

zarządzanie operatywne
- prognozowanie
- główny plan popytu
- główny harmonogram produkcji
- planowanie zdolności produkcyjnych
- planowanie materiałowe
- planowanie dystrybucji
- zarządzanie magazynami

marketing
- zarządzanie aktywnościami
- informacje o klientach i o potencjalnych klientach
- zarządzanie dokumentami marketingowymi
- planowanie kampanii

zasoby ludzkie
- administrowanie personelem
- administrowanie wynagrodzeniami

utrzymanie zdolności prod.
- obiekty techniczne
- serwis prewencyjny
- analiza techniczna i finansowa

jakość
- kontrola jakości
- analiza laboratoryjna
- śledzenie partii
- kontrola procesów statystycznych

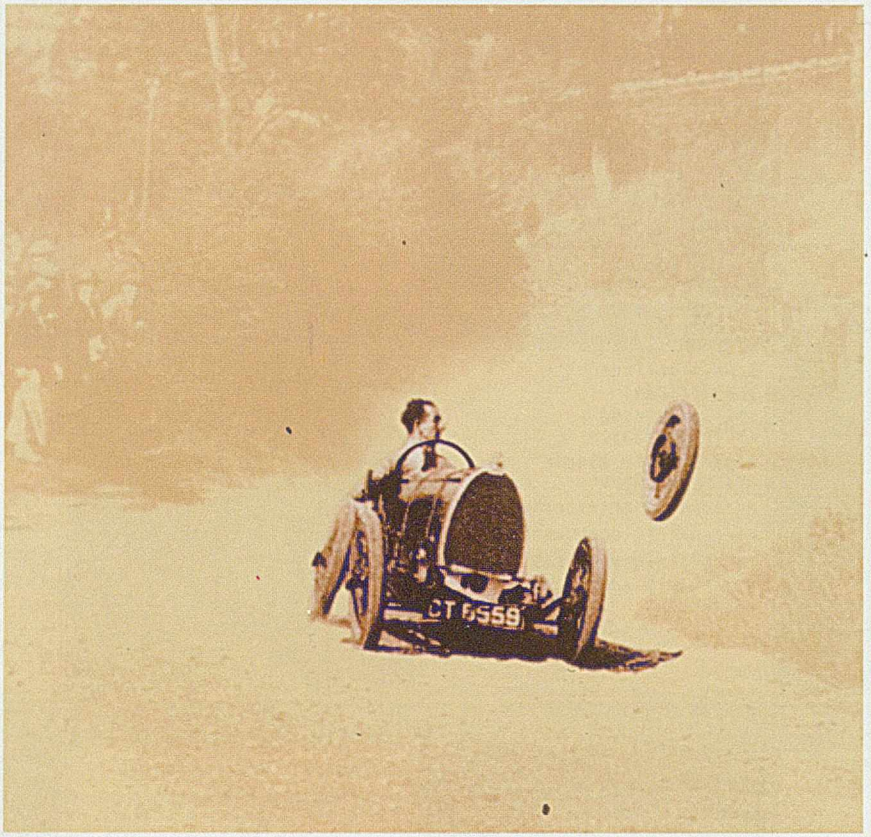
Intentia-VimeX

Z nami skomplikowane staje się proste

intentia-VimeX®

ul. Czerniakowska 81/83
00-718 Warszawa
tel./fax (22) 41 24 54, 41 55 61
e-mail: vimex@perytnet.pl

Porzuć stare technologie



IFS Applications - systemy wspierające zarządzanie przedsiębiorstwem - opierają się na Foundation1- zestawie narzędzi i metod zapewniających najwyższą jakość rozwoju i wdrażania systemów informatycznych. Foundation1 jest oparte na technologii relacyjnych baz danych Oracle, na trójwarstwowej architekturze client/server i na logice ukierunkowanej na obiekt. Zapewnia graficzny interfejs użytkownika i działa we wszystkich środowiskach Windows.

IFS Applications są przygotowane do zmian wynikających z przejścia w rok 2000 i pracują na wszystkich najważniejszych platformach sprzętowych takich jak: Windows NT Microsoft, różnorodne platformy UNIX np. AIX IBM oraz HP-UX Hewlett-Packard.

Dotychczas zaufali nam m.in. Stalprodukt, Nobiles, Centertel, Pilkington, Ciech, EC Jaworzno... Szczycimy się największą liczbą wdrożeń wśród 200 największych przedsiębiorstw w Polsce.

INDUSTRIAL & FINANCIAL SYSTEMS IFS

Set your business free.

IFS POLAND SP. Z.O.O. Warszawa Tel 022-608 46 00, Kraków Tel 012-422 50 15, Gdańsk Tel 058-45 29 64, Poznań Tel 022-608 46 00
www.ifsab.com info@wwa.ifs.com.pl